INDEPENDENT COMPONENT ANALYSIS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
in
Electronics and Communication
Engineering Department

By

P. SHIVA PRASASD

Department of E&C Engineering
National Institute of Technology
Rourkela
2007

INDEPENDENT COMPONENT ANALYSIS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
in
Electronics and Communication
Engineering Department

By

P. SHIVA PRASAD

Under the Guidance of

Prof. G.Panda

Department of E&C Engineering
National Institute of Technology
Rourkela

2007

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "INDEPENDENT COMPONENT ANALYSIS" submitted by Shri P Shiva Prasad. in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Electronics and comunication Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date: 03-05-2007

Prof G.Panda
Dept. of E&C Engineering
National Institute of Technology
Rourkela - 769008

Acknowledgment

I would like to express my sincere gratitude to Prof G.Panda for his invaluable guidance, cooperation and constant encouragement during the course of the project. I am grateful to Prof G.Panda, Head of the department, Electronics and Comunication Engineering for giving a lot of freedom, encouragement and guidance. I am also thankful to the Technical Staff of the DSP Laboratory, N.I.T. Rourkela for helping me during the experimental work.

P .Shiva Prasad. Date: 03rd May, 2007

10307008 .

Final Yr, Electronics and Comunication Engineering,

N.I.T Rourkela.

CONTENTS

INDEPENDENT COMPONENT ANALYSIS

A fundamental problem in neural network research, as well as in many other disciplines, is finding a suitable representation of multivariate data, i.e. random vectors. For reasons of computational and conceptual simplicity, the representation is often sought as a linear transformation of the original data. In other words, each component of the representation is a linear combination of the original variables. Well-known linear transformation methods include principal component analysis, factor analysis, and projection pursuit. Independent component analysis (ICA) is a recently developed method in which the goal is to find a linear representation of nongaussian data so that the components are statistically independent, or as independent as possible. Such a representation seems to capture the essential structure of the data in many applications, including feature extraction and signal separation.

In this report, we present the basic theory and applications of ICA, and our recent work on the subject , try to get a view of the principles underlying the working of independent component analysis. Next we will try to delve into various algorithms used in ICA analysis especially FastIca algorithm ,which is an efficient and a fast working algorithm

LIST OF FIGURES

Chapter 1

GENERAL INTRODUCTION

Independent component analysis (ICA) is a stastical and computational method for finding underlying factors or components from multivariate (multidimensional) statistical data. What distinguishes ICA from other methods is that it looks for components that are both *statistically independent*, and *nongaussian.* Here we briefly introduce the basic concepts, applications, and estimation principles of ICA.

ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed nongaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA.
The data analyzed by ICA could originate from many different kinds of application fields, including digital images, document databases, economic indicators and psychometric measurements.

1.1 BLIND SOURCE SEPARATION

1.1.1 Observing mixtures of unknown signals-

Consider a situation where there are a number of signals emitted by some physical objects or sources. These physical sources could be, for example, different brain areas emitting electric signals; people speaking in the same room, thus emitting speech signals; or mobile phones emitting their radio waves. Assume further that there are several sensors or receivers. These sensors are in different positions, so that each records a mixture of the original source signals with slightly different weights.

For the sake of simplicity of exposition, let us say there are three underlying source signals, and also three observed signals .Denoted by x1(t),x2(t),x3(t) the observed signals are the amplitudes of the observed signals at time t and by s1(t),s2(t),s3(t),the original signal .The xi(t) is the weighted sum of si(t),where the coefficients depend upon the distance from the source and the sensor.

$$x_1(t) = a_{11}s_1 + a_{12}s_2 + a_{13}s_3$$
$$x_2(t) = a_{21}s_1 + a_{22}s_2 + a_{23}s_3$$
$$x_3(t) = a_{31}s_1 + a_{32}s_2 + a_{33}s_3$$

The a_{ij} are constant coefficients that give the mixing weights. They are assumed unknown, since we cannot know the values of a_{ij} without knowing all the properties of the physical mixing system, which can be extremely difficult in general. The source signals a_{ij} are unknown as well, since the very problem is that we cannot record them directly.

What we would like to do is to find the original signals from the mixtures . x_1, x_2 and x_3 .This is the blind source separation (BSS) problem. Blind means that we know very little if anything about the original sources. We can safely assume that the mixing coefficients a_{ij} are different enough to make the matrix that they form invertible. Thus there exists a matrix W with coefficients w_{ij} , such that we can separate the s_i as

$$s_1(t) = w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$
$$s_2(t) = w_{21}x_1 + w_{22}x_2 + w_{23}x_3$$
$$s_3(t) = w_{31}x_1 + w_{32}x_2 + w_{33}x_3$$

Such a matrix W could be found as the inverse of the matrix that consists of the mixing coefficients a_{ij} in 1st Eq , if we knew those coefficients a_{ij} .

1.1.2 Source separation based on independence

The question now is: How can we estimate the coefficients w_{ij} .We want to obtain a general method that works in many different circumstances, and in fact provides one

answer to the very general problem that we started with: finding a good representation of multivariate data. Therefore, we use very general statistical properties. All we observe is the signals x_1, x_2 and x_3. we want to find a matrix W so that the representation is given by the original source signals s_1, s_2 and s_3.

A surprisingly simple solution to the problem can be found by considering just the statistical independence of the signals. In fact, if the signals are not gaussian, it is enough to determine the coefficients w_{ij}, so that the signals

$$y_1(t) = w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$
$$y_2(t) = w_{21}x_1 + w_{22}x_2 + w_{23}x_3$$
$$y_3(t) = w_{31}x_1 + w_{32}x_2 + w_{33}x_3$$

are statistically independent. If the signals y_1, y_2 and y_3 are independent, then they are equal to the original signals s_1, s_2 and s_3 (They could be multiplied by some scalar constants, though, but this has little significance).Using just this information on the statistical independence, we can in fact estimate the coefficient matrix W for the signals. What we obtain are the source signals . We see that from a data set that seemed to be just noise, we were able to estimate the original source signals, using an algorithm that used the information on the independence only. These estimated signals are indeed equal to those that were used in creating the mixtures.

We have now seen that the problem of blind source separation boils down to finding a linear representation in which the components are statistically independent. In practical situations, we cannot in general find a representation where the components are really independent, but we can at least find components that are as independent as possible.

Given a set of observations of random variables $x_1(t), x_2(t), x_3(t), ...x_n(t)$ where t is the time or sample index, assume that they are generated as a linear mixture of independent components:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = A \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} \qquad (1)$$

Where X is the received signal matrix , A is the unknown matrix. Independent component analysis now consists of estimating both the matrix A and the S , when we only observe $x_i(t)$ that we assumed here that the number of independent components $s_i(t)$ is equal to the number of observed variables; this is a simplifying assumption that is not completely necessary.

Alternatively ,we could define ICA as follows: find a linear transformation given by a matrix W as in , so that the random variables y_i ,i=1….n are as independent as possible. This formulation is not really very different from the previous one, since after estimating A its inverse gives W.

It can be shown that the problem is well-defined,that is, the model in (1)can be estimated if and only if the components si are nongaussian .This is a fundamental requirement that also explains the main difference between ICA and factor analysis, in which the nongaussianity of the data is not taken into account. In fact, ICA could be considered as nongaussian factor analysis, since in factor analysis, we are also modeling the data as linear mixtures of some underlying factors .

1.2 How to find the independent components

The independent components can be estimated from linear mixtures with no more assumptions than their independence. Now we will try to explain briefly why and how this is possible.

1.2.1 Uncorrelatedness is not enough

The first thing to note is that independence is a much stronger property than Uncorrelatedness. Considering the blind source separation problem, we could actually find many different uncorrelated representations of the signals that would not be independent and would not separate the sources. Uncorrelatedness in itself is not enough to separate the components. This is also the reason why principal component analysis

(PCA) or factor analysis cannot separate the signals: they give components that are uncorrelated, but little more.

1.2.2 Nonlinear decorrelation is the basic ICA method

One way of stating how independence is stronger than uncorrelatedness is to say that independence implies nonlinear uncorrelatedness :

If $s_1(t)$ and $s_2(t)$ are independent, then any nonlinear transformations g($s_1(t)$) and h($s_2(t)$) are uncorrelated (in the sense that their covariance is Independent Component Analysis zero). In contrast, for two random variables that are merely uncorrelated, such nonlinear transformations do not have zero covariance in general .Thus , we could attempt to perform ICA by a stronger form of decorrelation, by finding a representation where the y_i are uncorrelated even after some nonlinear transformations. This gives a simple principle of estimating the matrix W :

ICA estimation principle 1: Nonlinear decorrelation. Find the matrix W so that for any $i \neq j$, the components y_i and y_j are uncorrelated, and the transformed components g(y_i) and h(y_j) are uncorrelated, where g and h are some suitable nonlinear functions.

This is a valid approach to estimating ICA: If the nonlinearities are properly chosen, the method does find the independent components. Although this principle is very intuitive, it leaves open an important question:
How should the nonlinearities g and h be chosen? Answers to this question can be found be using principles from estimation theory and information theory. Estimation theory provides the most classic method of estimating any statistical model: the maximum likelihood method. Information theory provides exact measures of independence, such as mutual information. Using either one of these theories, we can determine the nonlinear functions g and h in a satisfactory way.

1.2.3 Independent components are the maximally nongaussian components

Another very intuitive and important principle of ICA estimation is maximum nongaussianity
. The idea is that according to the central limit theorem, sums of nongaussian random variables are closer to gaussian that the original ones. Therefore , if we take a linear combination $y = \sum b_i x_i$ of the observed mixture variables (which, because of the linear mixing model, is a linear combination of the independent components as well), this will be maximally nongaussian if it equals one of the independent components. This is because if it were a real mixture of two or more components, it would be closer to a gaussian distribution, due to the central limit theorem.
Thus, the principle can be stated as follows:

ICA estimation principle 2: Maximum nongaussianity. Find the local maxima of nongaussianity of a linear combination $y = \sum b_i x_i$ under the constraint that the variance of y is constant. Each local maximum gives one independent component

To measure nongaussianity in practice, we could use, for example, the *kurtosis*. Kurtosis is a higher order cumulant, which are some kind of generalizations of variance using higher order polynomials.

1.3 Independent component analysis

1.3.1 Motivation-

Imagine that you are in a room where two people are speaking simultaneously. You have two microphones, which you hold in different locations. The microphones give you two recorded time signals, which we could denote by x1(t) and x2(t), with x1 and x2 the amplitudes, and t the time index. Each of these recorded signals is a weighted sum of the speech signals emitted by the two speakers, which we denote by s1(t) and s2(t). We could express this as a linear equation:

$$x_1 = a_{11}s + a_{12}s_2$$
$$x_2 = a_{21}s_1 + a_{22}s_2 \qquad \text{eq.(1) \& (2)}$$

where $a_{11}, a_{12}, a_{21}, a_{22}$ are some parameters that depend on the distances of the microphones from the speakers. It would be very useful if you could now estimate the two original speech signals $s_1(t)$ and $s_2(t)$, using only the recorded signals $x_1(t)$ and $x_2(t)$. This is called the cocktail-party problem. For the time being, we omit any time delays or other extra factors from our simplified mixing model.As an illustration, consider the waveforms in Fig. 1 and Fig. 2. These are, of course, not realistic speech signals, but suffice for this illustration. The original speech signals could look something like those in Fig. 1 and the mixed signals could look like those in Fig. 2. The problem is to recover the data in Fig. 1 using only the data in Fig. 2. Actually, if we knew the parameters a_{ij}, we could solve the linear equation in (1) by classical methods. The point is, however, that if you don't know the a_{ij}, the problem is considerably more difficult. One approach to solving this problem would be to use some information on the statistical properties of the signals $s_i(t)$ to estimate the a_{ij}. Actually, and perhaps surprisingly, it turns out that it is enough to assume that $s_1(t)$ and $s_2(t)$, at each time instant t, are statistically independent. This is not an unrealistic assumption in many cases, and it need not be exactly true in practice. The recently developed technique of Independent Component Analysis, or ICA, can be used to estimate the a_{ij} based on the information of their independence, which allows us to separate the two original source signals $s_1(t)$ and $s_2(t)$ from their mixtures $x_1(t)$ and $x_2(t)$.Fig. 3 gives the two signals

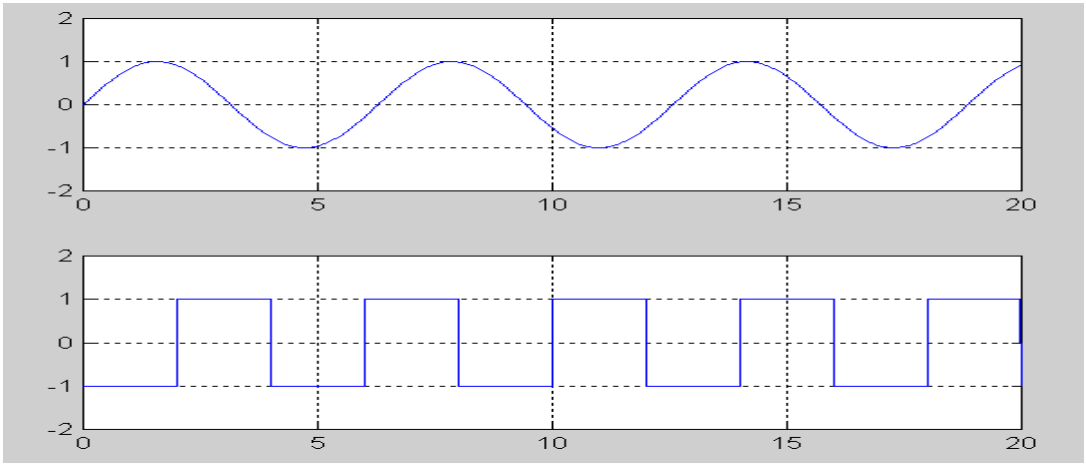


Fig . 1.1 Source signals

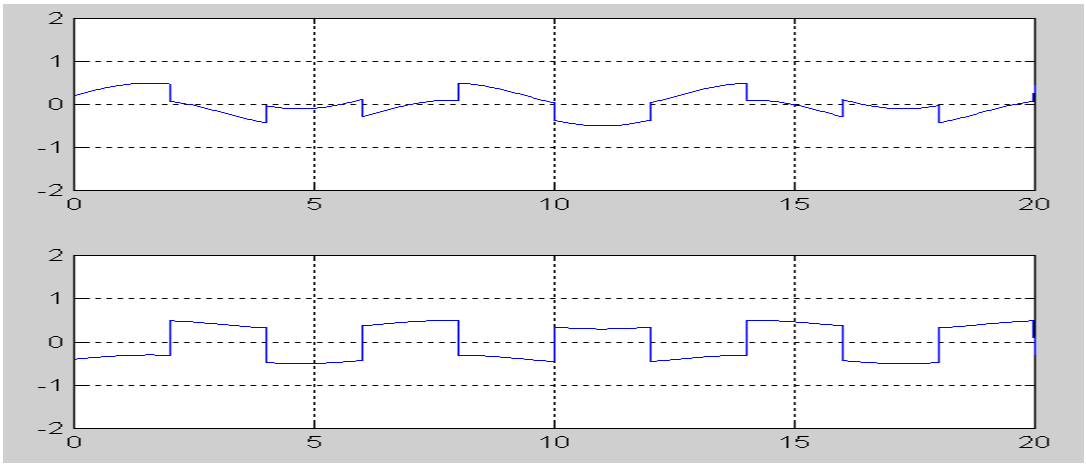


Fig 1.2 linearly mixed signals

estimated by the ICA method. As can be seen, these are very close to the original source signals (their signs are reversed, but this has no significance.)Independent component analysis was originally developed to deal with problems that are closely related to the cocktail-party problem. Since the recent increase of interest in ICA, it has become clear that this principle has a lot of other interesting applications as well.

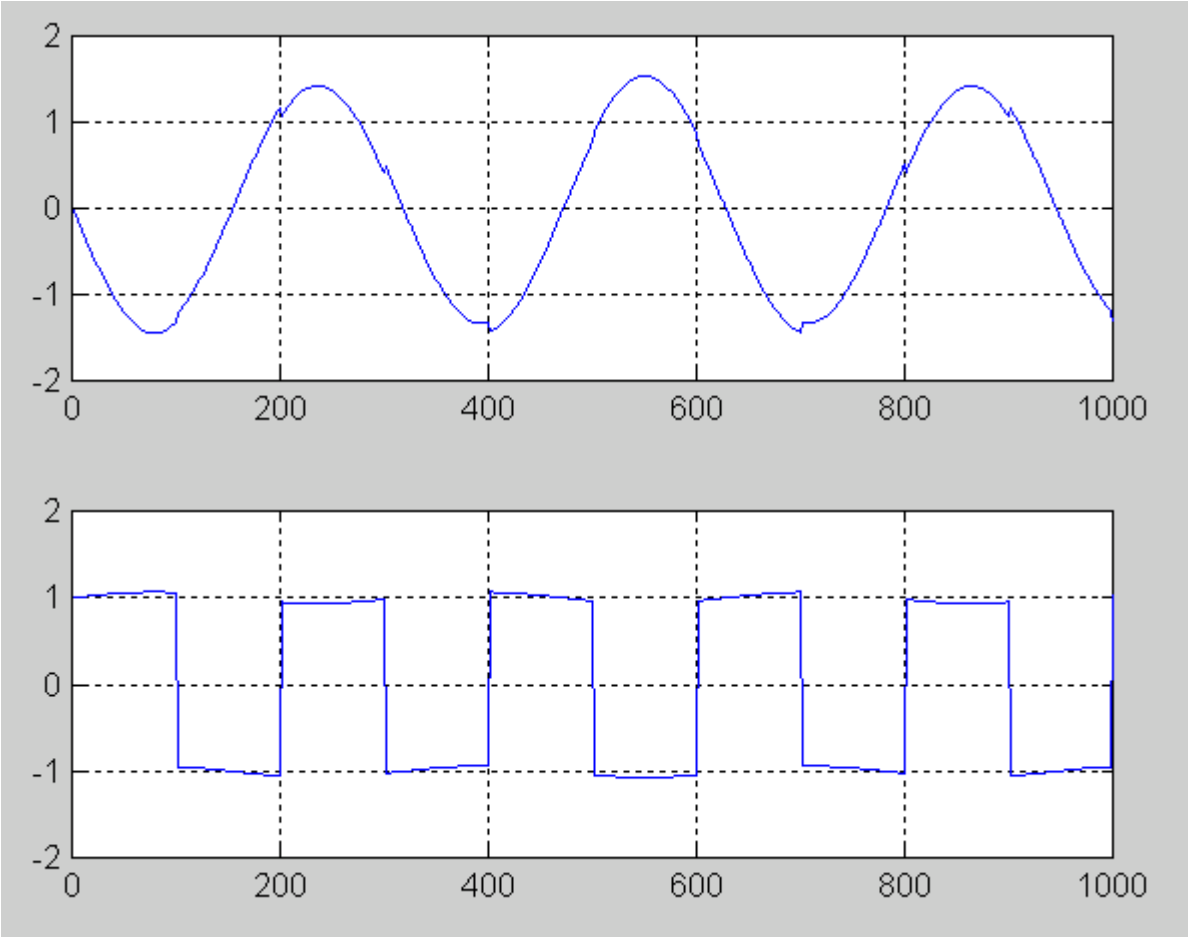


Fig . 1.3 Recovered signal

1.3.2 Definition of ICA-

To rigorously define ICA (Jutten and Hérault, 1991; Comon, 1994), we can use a statistical "latent variables"model. Assume that we observe n linear mixtures $x_1,...x_n$ of n independent components.

$x_j = a_{j1}s_1 + a_{j2}s_2 + ... + a_{jn}s_n$ for all j (3)

We have now dropped the time index t; in the ICA model, we assume that each mixture x_j as well as each independent component s_k is a random variable, instead of a proper time signal. The observed values $x_j(t)$, e.g.,the microphone signals in the cocktail party

problem, are then a sample of this random variable. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean: If this is not true, then the observable variables xi can always be centered by subtracting the sample mean, which makes the model zero-mean.

It is convenient to use vector-matrix notation instead of the sums like in the previous equation. Let us denote by x the random vector whose elements are the mixtures $x_1,...x_n$ and likewise by s the random vector with elements $s_1,...,s_n$. Let us denote by A the matrix with elements a_{ij}. Generally, bold lower case letters indicate vectors and bold upper-case letters denote matrices. All vectors are understood as column vectors; thus x^T, or the transpose of x, is a row vector. Using this vector-matrix notation, the above mixing model is written as

$$x = As \qquad (4)$$

Sometimes we need the columns of matrix A; denoting them by a_j the model can also be written as

$$x = \sum_{i=1}^{n} a_i s_i \qquad (5)$$

The statistical model in Eq. 4 is called independent component analysis, or ICA model. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components s_i. The independent omponents are latent variables, meaning that they cannot be directly observed.Also the mixing matrix is assumed to be unknown. All we observe is the random vector x, and we must estimate both A and s using it. This must be done under as general assumptions as possible. The starting point for ICA is the very simple assumption that the components si are statistically independent.. It will be seen below that we must also assume that the independent component must have nongaussian distributions. However, in the basic model we do not assume these distributions known (if they are known, the problem is considerably simplified.) For simplicity, we are also assuming that the unknown mixing matrix is square, but this assumption can be sometimes relaxed. Then, after estimating the matrix A, we can compute its inverse, say W, and obtain the independent

component simply by:

$$\mathbf{s} = \mathbf{W}\mathbf{x}. \tag{8}$$

ICA is very closely related to the method called blind source separation (BSS) or blind signal separation. A "source" means here an original signal, i.e. independent component, like the speaker in a cocktail party problem. "Blind" means that we no very little, if anything, on the mixing matrix, and make little assumptions on the source signals. ICA is one method, perhaps the most widely used, for performing blind source separation.

1.3.3 Independence

Definition and fundamental properties

To define the concept of independence, consider two scalar-valued random variables y_1 and y_2. Basically, the variables y_1 and y_2 are said to be independent if information on the value of y_1 does not give any information on the value of y_2, and vice versa. Above, we noted that this is the case with the variables s1, s2 but not with the mixture variables x_1, x_2.Technically, independence can be defined by the probability densities. Let us denote by p(y_1, y_2) the joint probability density function (pdf) of y_1 and y_2. Let us further denote by p1(y_1) the marginal pdf of y_1, i.e. the pdf of y_1 when it is considered alone:

$$p_1(y_1) = \int p(y_1, y_2) dy_2 \tag{9}$$

and similarly for y_2. Then we define that y_1 and y_2 are independent if and only if the joint pdf is factorizable in the following way:

$$p(y_1, y_2) = p_1(y_1) p_2(y_2) \tag{10}$$

This definition extends naturally for any number n of random variables, in which case the joint density must be a product of n terms.

The definition can be used to derive a most important property of independent random variables. Given two functions , h_1 and h_2 , we always have

$$E\{h_1(y_1)h_2(y_2)\} = E(h_1)E(h_2) \qquad (12)$$

Uncorrelated variables are only partly independent-

A weaker form of independence is uncorrelatedness. Two random variables y1 and y2 are said to be uncorrelated, if their covariance is zero:

$$E(y_1y_2) - E(y_1)E(y_2) = 0 \qquad (13)$$

If the variables are independent, they are uncorrelated, which follows directly from Eq. (11), taking $h_1(y_1) = y_1$ and $h_2(y_2) = y_2$

On the other hand, uncorrelatedness does not imply independence. For example, assume that (y_1 , y_2) are discrete valued and follow such a distribution that the pair are with probability 1/4 equal to any of the following values: (0,1), (0,−1), (1,0), (−1,0). Then y_1 and y_2 are uncorrelated, as can be simply calculated. On the other hand,

$$E(y_1^2y_2^2) = 0 \neq \frac{1}{4} = E(y_1^2)E(y_2^2) \qquad (14)$$

so the condition in Eq. (11) is violated, and the variables cannot be independent. Since independence implies uncorrelatedness, many ICA methods constrain the estimation procedure so that it always gives uncorrelated estimates of the independent components. This reduces the number of free parameters, and simplifies the problem.

Why Gaussian variables are forbidden

The fundamental restriction in ICA is that the independent components must be nongaussian for ICA to be possible.

To see why gaussian variables make ICA impossible, assume that the mixing matrix is orthogonal and the si are gaussian. Then x_1 and x_2 are gaussian, uncorrelated, and of unit variance. Their joint density is given by

$$p(x_1, x_2) = \frac{1}{2\pi} \exp(-\frac{x_1^2 + x_2^2}{2}) \tag{15}$$

This distribution is illustrated in Fig . The Figure shows that the density is completely symmetric. Therefore, it does not contain any information on the directions of the columns of the mixing matrix $\mathbf{A}$. This is why $\mathbf{A}$ cannot be estimated.

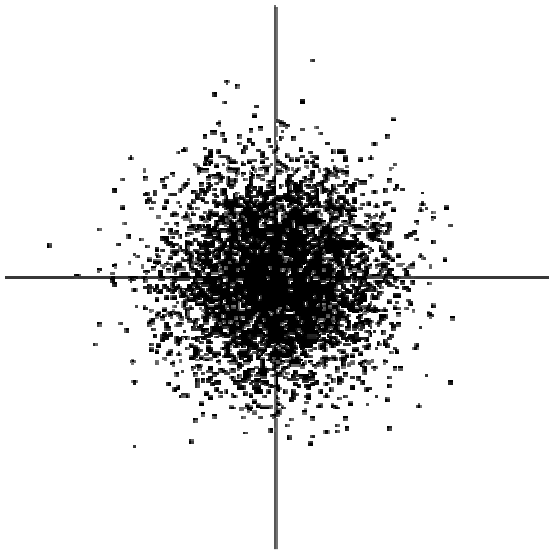


Fig 1.4 The multivariate distribution of two independent gaussian variables

1.3.4 Principles of ICA estimation

Intuitively speaking, the key to estimating the ICA model is nongaussianity. Actually, without nongaussianity the estimation is not possible at all . This is at the same

time probably the main reason for the rather late resurgence of ICA research: In most of classical statistical theory, random variables are assumed to have gaussian distributions, thus precluding any methods related to ICA. The Central Limit Theorem, a classical result in probability theory, tells that the distribution of a sum of independent random variables tends toward a gaussian distribution, under certain conditions. Thus, a sum of two independent random variables usually has a distribution that is closer to gaussian than any of the two original random variables. Let us now assume that the data vector x is distributed according to the ICA data model in Eq. 4, i.e. it is a mixture of independent components. For simplicity, let us assume in this section that all the independent components have identical distributions. To estimate one of the independent components, we consider a linear combination of the xi (see eq. 6); let us denote this by $y = w^T x = \sum_i w_i x_i$, where w is a vector to be determined . If w were one of the rows of the inverse of A, this linear combination would actually equal one of the independent components. The question is now: How could we use the Central Limit Theorem to determine w so that it would equal one of the rows of the inverse of A? In practice, we cannot determine such a w exactly, because we have no knowledge of matrix A, but we can find an estimator that gives a good approximation. To see how this leads to the basic principle of ICA estimation, let us make a change of variables, defining $z = A^T w$. Then we have $y = w^T x = w^T As = z^T s$. y is thus a linear combination of s_i , with weights given by z_i .Since a sum of even two independent random variables is more gaussian than the original variables, $z^T s$ is more gaussian than any of the s_i and becomes least gaussian when it in fact equals one of the s_i . In this case, obviously only one of the elements z_i of z is nonzero. (Note that the z_i were here assumed to have identical distributions.)

Therefore, we could take as w a vector that maximizes the nongaussianity of $w^T x$. Such a vector wouldnecessarily correspond (in the transformed coordinate system) to a z which has only one nonzero component. This means that $w^T x = z^T s$ equals one of the independent components! Maximizing the nongaussianity of $w^T x$ thus gives us one of the independent components. In fact, the optimization landscape for nongaussianity in the

n-dimensional space of vectors w has 2n local maxima, two for each independent component, corresponding to si and −si (recall that the independent components can be estimated only up to a multiplicative sign). To find several independent components, we need to find all these local maxima. This is not difficult, because the different independent components are uncorrelated: We can always constrain the search to the space that gives estimates uncorrelated with the previous ones. This corresponds to orthogonalization in a suitably transformed (i.e. whitened) space.

1.3.5 Measures of nongaussianity

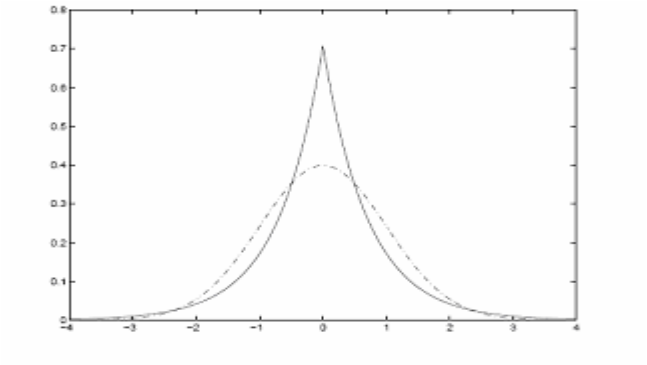
To use nongaussianity in ICA estimation, we must have a quantitative measure of nongaussianity of a random variable, say *y*. To simplify things, let us assume that *y* is centered (zero-mean) and has variance equal to one.
Actually, one of the functions of preprocessing in ICA algorithms, to be covered in Section 5, is to make this simplification possible.

1.3.5.1 Kurtosis

The classical measure of nongaussianity is kurtosis or the fourth-order cumulant. The kurtosis of *y* is classically defined by

$$\text{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$$

Actually, since we assumed that *y* is of unit variance, the right-hand side simplifies to $E\{y^4\}-3$. This shows that kurtosis is simply a normalized version of the fourth moment $E\{y^4\}$. For a gaussian *y*, the fourth moment equals $3(E\{y^2\})^2$. Thus, kurtosis is zero for a gaussian random variable. For most (but not quite all) nongaussian random variables, kurtosis is nonzero.



. **Figure:1.5** The density function of laplace distribution .Which is a typically a supergaussian distribution. The dashed line represents gaussian density. Both the densities are normalized

Kurtosis can be both positive and negative. Random variables that have a negative kurtosis are called subgaussian, and those with positive kurtosis are called supergaussian. In statistical literature, the corresponding expressions platykurtic and leptokurtic are also used. Supergaussian random variables have typically a "spiky" pdf with heavy tails, i.e. the pdf is relatively large at zero and at large values of the variable, while being small for intermediate values. A typical example is the Laplace distribution, whose pdf (normalized to unit variance) is given by

$$p(y) = \frac{1}{\sqrt{2}} \exp(\sqrt{2}|y|)$$

This pdf is illustrated in Fig. 1.3.1. Subgaussian random variables, on the other hand, have typically a "flat" pdf, which is rather constant near zero, and very small for larger values of the variable. A typical example is the uniform distribution in above equation.
Typically non gaussianity is measured by the absolute value of kurtosis. The square of kurtosis can also be used. These are zero for a gaussian variable, and greater than zero for most nongaussian random variables. There are nongaussian random variables that have zero kurtosis, but they can be considered as very rare.

Kurtosis, or rather its absolute value, has been widely used as a measure of nongaussianity in ICA and related fields. The main reason is its simplicity, both computational and theoretical. Computationally, kurtosis can be estimated simply by using the fourth moment of the sample data. Theoretical analysis is simplified because of the following linearity property: If x_1 and x_2 are two independent random variables, it holds

$$\text{kurt}(x_1 + x_2) = \text{kurt}(x_1) + \text{kurt}(x_2)$$

$$\text{kurt}(\alpha x_1) = \alpha^4 \text{kurt}(x_1)$$

These properties can be easily proven using the definition. To illustrate in a simple example what the optimization landscape for kurtosis looks like, and how independent components could be found by kurtosis minimization or maximization, let us look at a 2-dimensional model $\mathbf{x} = \mathbf{As}$. Assume that the independent components s_1, s_2 have kurtosis values Kurt(s_1), Kurt(s_2), respectively, both different from zero. Remember that we assumed that they have unit variances. We seek for one of the independent components as $y = \mathbf{w}^T \mathbf{x}$.

Let us again make the transformation $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then we have $y = \mathbf{w}^T \mathbf{x} = \mathbf{w}^T\mathbf{As} = \mathbf{z}^T \mathbf{s} = z_1s_1 + z_2s_2$. In practice we would start from some weight vector $\mathbf{w}$, compute the direction in which the kurtosis of $y = \mathbf{w}^T \mathbf{x}$ is growing most strongly (if kurtosis is positive) or decreasing most strongly (if kurtosis is negative) based on the available sample $\mathbf{x}(1)$, ...,$\mathbf{x}(T)$ of mixture vector $\mathbf{x}$, and use a gradient method or one of their extensions for finding a new vector $\mathbf{w}$. The example can be generalized to arbitrary dimensions, showing that kurtosis can theoretically be used as an optimization criterion for the ICA problem.

However, kurtosis has also some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is that kurtosis can be very sensitive to outliers (Huber, 1985). Its value may depend on only a few observations in

the tails of the distribution, which may be erroneous or irrelevant observations. In other words, kurtosis is not a robust measure of nongaussianity.

Thus, other measures of nongaussianity might be better than kurtosis in some situations. Below we shall consider negentropy whose properties are rather opposite to those of kurtosis, and finally introduce approximations of negentropy that more or less combine the good properties of both measures.

Mathematically the simplest one-unit contrast functions are provided by higher-order cumulants like kurtosis . Denote by $\mathbf{x}$ the observed data vector, assumed to follow the ICA data model. Now, let us search for a linear combination of the observations x_i, saytrans(W)*x, such that its kurtosis is maximized or minimized. Obviously, this optimization problem is meaningful only if $\mathbf{w}$ is somehow bounded; let us. Assume

$$E\{(\mathbf{w}^T\mathbf{x})^2\} = 1$$

Using the (unknown) mixing matrix $\mathbf{A}$, let us define $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then, using the data model $\mathbf{x} = \mathbf{As}$ one obtains

$$E\{(\mathbf{w}^T\mathbf{x})^2\} = \mathbf{w}^T\mathbf{A}\mathbf{A}^T\mathbf{w} = \|\mathbf{z}\|^2 = 1$$

$$\text{kurt}\,(\mathbf{w}^T\mathbf{x}) = \text{kurt}\,(\mathbf{w}^T\mathbf{As}) = \text{kurt}\,(\mathbf{z}^T\mathbf{s}) = \sum_{i=1}^{m} z_i^4\,\text{kurt}\,(s_i).$$

Under the constraint mag(square of z) is equal to 1, the function has a number of local minima and maxima. To make the argument clearer, let us assume for the moment that in the mixture there is at least one independent component s_j whose kurtosis is negative, and at least one whose kurtosis is positive. Then, as was shown in, the external points of above equation are the canonical base vectors

$$\mathbf{z} = \pm \mathbf{e}_j$$

i.e., vectors whose all components are zero except one component which is mod(1). The corresponding weight vectors are

$$\mathbf{w} = \pm (\mathbf{A}^{-1})^T \mathbf{e}_j$$

, i.e., the rows of the inverse of the mixing matrix $\mathbf{A}$, up to a multiplicative sign. So, by minimizing or maximizing the kurtosis in above equation. Under the given constraint, one obtains one of the independent components as

$$\mathbf{w}^T \mathbf{x} = \pm s_j$$

These two optimization modes can also be combined into a single one, because the independent components correspond always to maxima of the *modulus* of the kurtosis.

Kurtosis has been widely used for one-unit ICA as well as for projection pursuit. The mathematical simplicity of the cumulants, and especially the possibility of proving global convergence results contributed largely to the popularity of cumulant-based (one-unit) contrast functions in ICA, projection pursuit and related fields. However, it has been shown, for example that kurtosis often provides a rather poor objective function for the estimation of ICA, if the statistical properties of the resulting estimators are considered.

Note that despite the fact that there is no noise in the ICA model, neither the independent components nor the mixing matrix can be computed accurately because the independent components s_i are random variables, and, in practice, one only has a finite sample of $\mathbf{x}$. Therefore, the statistical properties of the estimators of $\mathbf{A}$and the realizations of $\mathbf{s}$can be analyzed just as the properties of any estimator. Such an analysis was conducted in, and the results show that in terms of robustness and asymptotic variance, the cumulant-based estimators tend to be far from optima. Intuitively, there are two main reasons for this. Firstly, higher-order cumulants measure mainly the tails of a distribution, and are largely unaffected by structure in the middle of the distribution. Secondly, estimators of higher-order cumulants are highly sensitive to outliers. Their value may depend on only a few observations in the tails of the distribution, which may be outliers.

1.3.5.2 Negentropy

A most natural information-theoretic one-unit contrast function is negentropy. one is tempted to conclude that the independent components correspond to directions in which the differential entropy of trans(w)*x is minimized. This turns out to be roughly the case. However, a modification has to be made, since differential entropy is not invariant for scale transformations. To obtain a linearly invariant version of entropy, one defines the negentropy *J* as follows

Where y(gaussian) is a Gaussian random vector of the same covariance matrix as 'y'. Negentropy, or negative normalized entropy, is always non-negative, and is zero if and only if 'y' has a Gaussian distribution.

The usefulness of this definition can be seen when mutual information is expressed using Negentropy, giving

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i)$$

Because Negentropy is invariant for linear transformations, it is now obvious that finding maximum Negentropy directions, i.e., directions where the elements of the sum $J(y_i)$are maximized, is equivalent to finding a representation in which mutual information is minimized. The use of Negentropy shows clearly the connection between ICA and projection pursuit. Using differential entropy as a projection pursuit index, as has been suggested, amounts to finding directions in which Negentropy is maximized.

Unfortunately, the reservations made with respect to mutual information are also valid here. The estimation of negentropy is difficult, and therefore this contrast function remains mainly a theoretical one. As in the multi-unit case, negentropy can be approximated by higher-order cumulants, for example as follows

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} \text{kurt}(y)^2$$

where k(y) is the *i*-th order cumulant of *y*. The random variable *y* is assumed to be of zero mean and unit variance. However, the validity of such approximations may be rather limited. It was argued that cumulant-based approximations of negentropy are inaccurate, and in many cases too sensitive to outliers. New approximations of negentropy were therefore introduced. In the simplest case, these new approximations are of the form:

where *G* is practically any non-quadratic function, *c* is an irrelevant constant, and ν is a Gaussian variable of zero mean and unit variance (i.e., standardized). For the practical choice of *G*, see below. These approximations were shown to be better than the cumulant-based ones in several respects.

Actually, the two approximations of negentropy discussed above are interesting as one-unit contrast functions in their own right, as will be discussed next.

Negentropy is based on the information theoretic quantity of (differential) entropy. Entropy is the basic concept of information theory. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy. More rigorously, entropy is closely related to the coding length of the random variable, in fact, under some simplifying assumptions, entropy *is* the coding length of the random variable. For introductions on information theory, see e.g. (Cover and Thomas, 1991; Papoulis, 1991).

Entropy *H* is defined for a discrete random variable *Y* as

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i)$$

Where the a_i are the possible values of Y. This very well-known definition can be generalized for continuous-valued random variables and vectors, in which case it is often called differential entropy. The differential entropy H of a random vector $\mathbf{y}$ with density $f(\mathbf{y})$ is defined as (Cover and Thomas, 1991; Papoulis, 1991):

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) \mathrm{d}\mathbf{y}.$$

A fundamental result of information theory is that *a gaussian variable has the largest entropy among all random variables of equal variance*. For a proof, see e.g. (Cover and Thomas, 1991; Papoulis, 1991). This means that entropy could be used as a measure of nongaussianity. In fact, this shows that the gaussian distribution is the "most random" or the least structured of all distributions. Entropy is small for distributions that are clearly concentrated on certain values, i.e., when the variable is clearly clustered, or has a pdf that is very "spiky".

To obtain a measure of nongaussianity that is zero for a gaussian variable and always nonnegative, one often uses a slightly modified version of the definition of differential entropy, called negentropy. Negentropy J is defined as follows

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

where $\mathbf{y}_{gauss}$ is a Gaussian random variable of the same covariance matrix as $\mathbf{y}$. Due to the above-mentioned properties, negentropy is always non-negative, and it is zero if and only if $\mathbf{y}$ has a Gaussian distribution. Negentropy has the additional interesting property that it is invariant for invertible linear transformations.

The advantage of using negentropy, or, equivalently, differential entropy, as a measure of nongaussianity is that it is well justified by statistical theory. In fact, negentropy is in some sense the optimal estimator of nongaussianity, as far as statistical properties are concerned. The problem in using negentropy is, however, that it is computationally very difficult. Estimating negentropy using the definition would require an estimate (possibly nonparametric) of the pdf. Therefore, simpler approximations of negentropy are very useful, as will be discussed next.

1.3.5.3 Approximations of negentropy

The estimation of negentropy is difficult, as mentioned above, and therefore this contrast function remains mainly a theoretical one. In practice, some approximation has to be used. Here we introduce approximations that have very promising properties, and which will be used in the following to derive an efficient method for ICA .The classical method of approximating negentropy is using higher-order moments, for example as follows

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} \mathrm{kurt}(y)^2$$

The random variable *y* is assumed to be of zero mean and unit variance. However, the validity of such approximations may be rather limited. In particular, these approximations suffer from the no robustness encountered with kurtosis.

To avoid the problems encountered with the preceding approximations of negentropy, new approximations were developed in (Hyvärinen, 1998b). These approximations were based on the maximum-entropy principle. In general we obtain the following approximation

$$J(y) \approx \sum_{i=1}^{p} k_i [E\{G_i(y)\} - E\{G_i(\nu)\}]^2$$

where *ki* are some positive constants, and n is a Gaussian variable of zero mean and unit variance (i.e., standardized).The variable *y* is assumed to be of zero mean and unit variance, and the functions *Gi* are some nonquadratic functions (Hyvärinen, 1998b). Note that even in cases where this approximation is not very accurate, can be used to construct a measure of nongaussianity that is consistent in the sense that it is always non-negative, and equal to zero if *y* has a Gaussian distribution. In the case where we use only one nonquadratic function *G*, the approximation becomes

For practically any non-quadratic function *G*. This is clearly a generalization of the moment-based approximation in above equ, if *y* is symmetric. Indeed, taking

$G(y)=y4$, one then obtains exactly above equ, i.e. a kurtosis-based approximation. But the point here is that by choosing *G* wisely, one obtains approximations of negentropy that are much better than the one given by above equ. In particular, choosing *G* that does not grow too fast, one obtains more robust estimators. The following choices of *G* have proved very useful:

Where $1<a<2$ is some suitable constant.. Thus we obtain approximations of negentropy that give a very good compromise between the properties of the two classical nongaussianity measures given by kurtosis and negentropy. They are conceptually simple, fast to compute, yet have appealing statistical properties, especially robustness. Therefore, we shall use these contrast functions in our ICA methods. Since kurtosis can be expressed in this same framework, it can still be used by our ICA methods. A practical algorithm based on these contrast function will be presented in Section coming section.

To avoid the problems encountered with the preceding objective functions, new one-unit contrast functions for ICA were developed . Such contrast functions try to combine the positive properties of the preceding contrast functions, i.e. have statistically appealing properties (in contrast to cumulants), require no prior knowledge of the densities of the independent components (in contrast to basic maximum likelihood estimation), allow a simple algorithmic implementation (in contrast to maximum likelihood approach with simultaneous estimation of the densities), and be simple to analyze (in contrast to non-linear cross-correlation and non-linear PCA approaches). The generalized contrast functions, which can be considered generalizations of kurtosis, seem to fulfill these requirements.

To begin with, note that one intuitive interpretation of contrast functions is that they are measures of non-normality. A family of such measures of non-normality could be constructed using practically any functions *G*, and considering the difference of the expectation of *G* for the actual data and the expectation of *G* for Gaussian data. In other words, we can define a contrast function *t*hat measures the non-normality of a zero-mean random variable *u*sing any even, non-quadratic, sufficiently smooth function *G* as follows

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} \mathrm{kurt}(y)^2$$

Where ν is a standardized Gaussian random variable, y is assumed to be normalized to unit variance, and the exponent p=1,2 typically. The subscripts denote expectation with respect to y and ν. (The notation J_G should not be confused with the notation for negentropy, J.)

Clearly, J_G can be considered a generalization of (the modulus of) kurtosis. For $G(y)=y^4$, J_G becomes simply the modulus of kurtosis of y. Note that G must not be quadratic, because then J_G would be trivially zero for all distributions. Thus, it seems plausible that J_G in a paper could be a contrast function in the same way as kurtosis. The fact that J_G is indeed a contrast function in a suitable sense (locally). In fact, for p=2, J_G coincides with the approximation of negentropy given in a paper.

The finite-sample statistical properties of the estimators based on optimizing such a general contrast function were analyzed. It was found that for a suitable choice of G, the statistical properties of the estimator (asymptotic variance and robustness) are considerably better than the properties of the cumulant-based estimators. The following choices of G were proposed: where a1,a2<1are some suitable constants. In the lack of precise knowledge on the distributions of the independent components or on the outliers, these two functions seem to approximate reasonably well the optimal contrast function in most cases. Experimentally, it was found that especially the values 1<a1<2, a2=1for the constants give good approximations. One reason for this is that G_1 above corresponds to the log-density of a super-gaussian distribution, and is therefore closely related to maximum likelihood estimation.

1.3.5.4 Minimization of Mutual Information

Another approach for ICA estimation, inspired by information theory, is minimization of mutual information. We will explain this approach here, and show that it

leads to the same principle of finding most nongaussian directions as was described above. In particular, this approach gives a rigorous justification for the heuristic rinciples used above.

1.3.5.5 Mutual Information

Using the concept of differential entropy, we define the mutual information *I* between *m* (scalar) random variables,

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}).$$

Mutual information is a natural measure of the dependence between random variables. In fact, it is equivalent to the well-known Kullback-Leibler divergence between the joint density $f(\mathbf{y})$ and the product of its marginal densities; a very natural measure for independence. It is always non-negative, and zero if and only if the variables are statistically independent. Thus, mutual information takes into account the whole dependence structure of the variables, and not only the covariance, like PCA and related methods.

Mutual information can be interpreted by using the interpretation of entropy as code length. The terms $H(yi)$ give the lengths of codes for the *yi* when these are coded separately, and $H(\mathbf{y})$ gives the code length when **y** is coded as a random vector, i.e. all the components are coded in the same code. Mutual information thus shows what code length reduction is obtained by coding the whole vector instead of the separate components. In general, better codes can be obtained by coding the whole vector. However, if the *yi* are independent, they give no information on each other, and one could just as well code the variables separately without increasing code length. An important property of mutual information (Papoulis, 1991; Cover and Thomas, 1991) is that we have for an invertible linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$:

$$I(y_1, y_2, \ldots, y_n) = \sum_i H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{W}|$$

$$\det \mathbf{I} = 1 = (\det \mathbf{W} E\{\mathbf{x}\mathbf{x}^T\} \mathbf{W}^T) = (\det \mathbf{W})(\det E\{\mathbf{x}\mathbf{x}^T\})(\det \mathbf{W}^T),$$

$$I(y_1, y_2, \ldots, y_n) = C - \sum_i J(y_i).$$

1.3.5.6 Defining ICA by Mutual Information

Since mutual information is the natural information-theoretic measure of the independence of random variables, we could use it as the criterion for finding the ICA transform. In this approach that is an alternative to the model estimation approach, we define the ICA of a random vector **x** as an invertible transformation , where the Matrix **W** is determined so that the mutual information of the transformed components *si* is minimized. It is now obvious from that finding an invertible transformation. that minimizes the mutual information is roughly equivalent to *finding directions in which the negentropy is maximized*. More precisely, it is roughly equivalent to finding 1-D subspaces such that the projections in those subspaces have maximum negentropy. Rigorously, speaking, above equ shows that ICA estimation by minimization of mutual information is equivalent to maximizing the sum of nongaussianities of the estimates, when the *estimates are constrained to be uncorrelated*. The constraint of uncorrelatedness is in fact not necessary, but simplifies the computations considerably, as one can then use the simpler form in above equ instead of the more complicated form . Thus, we see that the formulation of ICA as minimization of mutual information gives another rigorous justification of our more heuristically introduced idea of finding maximally nongaussian directions.

1.3.5.7 Maximum Likelihood Estimation

1.The likelihood

A very popular approach for estimating the ICA model is maximum likelihood estimation, which is closely connected to the infomax principle. Here we discuss this approach, and show that it is essentially equivalent to minimization of mutual information. It is possible to formulate directly the likelihood in the noise-free ICA

model, which was done in (Phamet al., 1992), and then estimate the model by a maximum likelihood method. Denoting by= (**w**1, ...,**w***n*)*T* the matrix **A**−1, the log-likelihood takes the form (Pham et al., 1992):

$$L = \sum_{t=1}^{T} \sum_{i=1}^{n} \log f_i(\mathbf{w}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{W}|$$

Where the *fi* are the density functions of the *si* (here assumed to be known), and the **x**(*t*), *t* = 1, ...,*T* are the realizations of **x**. The term log|det**W**| in the likelihood comes from the classic rule for (linearly) transforming random variables and their densities (Papoulis, 1991): In general, for any random vector **x** with density *px* and for any matrix

2. Connection to mutual information

To see the connection between likelihood and mutual information, consider the expectation of the log-likelihood:

$$\frac{1}{T} E\{L\} = \sum_{i=1}^{n} E\{\log f_i(\mathbf{w}_i^T \mathbf{x})\} + \log |\det \mathbf{W}|.$$

Actually, if the *fi* were equal to the actual distributions of **w***Ti* **x**, the first term would be equal to −å*iH*(**w***Ti* **x**). Thus the likelihood would be equal, up to an additive constant, to the negative of mutual information as given in Actually, in practice the connection is even stronger. This is because in practice we don't know the distributions of the independent components. A reasonable approach would be to estimate the density of **w***Ti* **x** as part of the ML estimation method, and use this as an approximation of the density of *si*. In this case, likelihood and mutual information are, for all practical purposes, equivalent. Nevertheless, there is a small difference that may be very important in practice. The problem with maximum likelihood estimation is that the densities *fi* must be estimated correctly. They need not be estimated with any great precision: in fact it is

enough to estimate whether they are sub- or supergaussian. In many cases, in fact, we have enough prior knowledge on the independent components, and we don't need to estimate their nature from the data. In any case, if the information on the nature of the independent components is not correct, ML estimation will give completely wrong results. Some care must be taken with ML estimation, therefore. In contrast, using reasonable measures of nongaussianity, this problem does not usually arise.

1.3.6. Preprocessing of the data

In the preceding section, we discussed the statistical principles underlying ICA methods. Practical algorithms based on these principles will be discussed in the next section. However, before applying an ICA algorithm on the data, it is usually very useful to do some preprocessing. In this section, we discuss some preprocessing techniques that make the problem of ICA estimation simpler and better conditioned

1.3.6.1 Centering

The most basic and necessary preprocessing is to center x, i.e. subtract its mean vector m = $E\{x\}$ so as to make x a zero-mean variable. This implies that s is zero-mean as well, as can be seen by taking expectations on both sides basic sensor output equations.
This preprocessing is made solely to simplify the ICA algorithms: It does not mean that the mean could not be estimated. After estimating the mixing matrix A with centered data, we can complete the estimation by adding the mean vector of s back to the centered estimates of s. The mean vector of s is given by $A^{-1}m$, where m is the mean that was subtracted in the preprocessing.

1.3.6.2 Whitening

Another useful preprocessing strategy in ICA is to first whiten the observed variables. This means that before the application of the ICA algorithm (and after centering), we transform the observed vector x *linearly* so that we obtain a new vector ˜x which is white, i.e. its components are uncorrelated and their variances equal unity. In other words, the covariance matrix of ˜x equals the identity matrix:

The whitening transformation is always possible. One popular method for whitening is to use the eigen-value decomposition (EVD) of the covariance matrix $E\{xx^T\}=EDE^T$, where E is the orthogonal matrix of eigenvectors of $E\{xx^T\}$ and D is the diagonal matrix of its eigenvalues, D= diag(d_1, ...,d_n). Note that $E\{xx^T\}$ can be estimated in a standard way from the available sample x(1), ...,x(T). Whitening can now be done by

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x}$$

where the matrix $D^{-1/2}$is computed by a simple component-wise operation as $D^{-1/2}$ = diag($d^{-1/2}$ 1 , ...,$d^{-1/2}$ n). It is easy to check that now $E\{\tilde{x}\tilde{x}^T\}$ = I.
Whitening transforms the mixing matrix into a new one, ˜A. We have from (4) and (35):

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{A}\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s}$$

The utility of whitening resides in the fact that the new mixing matrix ˜A is orthogonal. This can be seen from

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \tilde{\mathbf{A}}E\{\mathbf{s}\mathbf{s}^T\}\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I}.$$

Here we see that whitening reduces the number of parameters to be estimated. Instead of having to estimate the n_2 parameters that are the elements of the original matrix A, we only need to estimate the new, orthogonal mixing matrix ˜A. An orthogonal matrix contains $n(n-1)/2$ degrees of freedom. For example, in two dimensions, an orthogonal transformation is determined by a single angle parameter. In larger dimensions, an orthogonal matrix contains only about half of the number of parameters of an arbitrary matrix. Thus one can say that whitening solves half of the problem of ICA. Because whitening is a very simple and standard procedure, much simpler than any ICA algorithms, it is a good idea to reduce the complexity of the problem this way.

It may also be quite useful to reduce the dimension of the data at the same time as we do the whitening. Then we look at the eigen values d_j of $E\{xx^T\}$ and discard those that are too small, as is often done in the statistical technique of principal component analysis.

Chapter 2

LITERTURE REVIEW

2.1 Brief History-

The technique of ICA, although not yet the name, was introduced in the early 1980s by J. H´erault , C. Jutten , and B. Ans. As recently reviewed by Jutten the problem first came up in 1982 in a neurophysiological setting. . In a simplified model of muscle coding in muscle contraction, the outputs x1(t) and x2(t) were two types of sensory signals measuring muscle contraction and s1(t) and s2(t) were the angular position and velocity of a moving joint. Then it is not unreasonable to assume that the ICA model holds between these signals. The nervous system must be somehow able to infer the position and velocity signals s1(t) and s2(t) from the measured responses x1(t) and x2(t). One possibility for this is to learn the inverse model using the nonlinear decorrelation principle in a simple neural network. H´erault and Jutten proposed a specific feedback circuit solve the problem.

All through the 1980s, ICA was mostly known among French researchers, with limited influence internationally. The few ICA presentations in international neural network conferences in the mid1980s were largely buried under the deluge of interest in back propagation , Hopfield networks, and Kohonen's Self Organizing Map (SOM) , which were actively propagated in those times.

Another related field was higher order spectral analysis, on which the first international workshop was organized in 1989. In this workshop, early papers on ICA by J.F. Cardoso and P. Comon were given. Cardoso used algebraic methods, especially higher order cumulant tensors, which eventually led to the JADE algorithm. The use of fourth order cumulants has been earlier proposed by J.L. Lacoume. A good source with historical accounts and a more complete list of references is. In signal processing, there had been earlier approaches in the related problem of blind signal deconvolution . In particular, the results used in multichannel blind deconvolutions are very similar to ICA techniques. The work of the scientists in the 1980's was extended by, among others, A. Cichocki and R. Unbehauen, who were the first to propose one of the presently most popular ICA

algorithms . The “nonlinear PCA” approach was introduced by the present authors.However, until the mid1990s, ICA remained a rather small and narrow research effort. Several algorithms were proposed that worked, usually in somewhat restricted problems, but it was not until later that the rigorous connections of these to statistical optimization criteria were exposed. ICA attained wider attention and growing interest after A.J. Bell and T.J. Sejnowski published their approach based on the infomax principle in the mid90’s.This algorithm was further refined by S.I. Amari and his coworkers using the natural gradient , and its fundamental connections to maximum likelihood estimation, as well as to the CichockiUnbehauen algorithm , were established. A couple of years later, they presented the fixed point or FastICA algorithm, which has contributed to the application of ICA to large scale problems due to its computational efficiency.

2.2 Algorithms for ICA

>>Jutten-Hérault algorithm

>>Non-linear decorrelation algorithms

>>Algorithms for maximum likelihood or infomax estimation

>>Non-linear PCA algorithms

>> Neural one-unit learning rules

>> Other neural (adaptive) algorithms

>> The FastICA algorithm

>>Tensor-based algorithms

>>Weighted covariance methods

Chapter 3

EXPERIMENTAL PROGRAM

3.1 Algorithms for ICA

>>Jutten-Hérault algorithm

>>Non-linear decorrelation algorithms

>>Algorithms for maximum likelihood or infomax estimation

>>Non-linear PCA algorithms

>> Neural one-unit learning rules

>> Other neural (adaptive) algorithms

>> The FastICA algorithm

>>Tensor-based algorithms

>>Weighted covariance methods

3.2 Choice of algorithm

To summarize, the choice of the ICA algorithm is basically a choice between adaptive and batch-mode (block) algorithms.

In the adaptive case, the algorithms are obtained by stochastic gradient methods. In the case where all the independent components are estimated at the same time, the most popular algorithm in this category is natural gradient ascent of likelihood, or related contrast functions, like infomax the one-unit case, straightforward stochastic gradient methods give adaptive algorithms that maximize negentropy or its approximations.

In the case where the computations are made in batch-mode, much more efficient algorithms are available. The tensor-based methods are efficient in small dimensions, but they cannot be used in larger dimensions. The FastICA algorithm, based on a fixed-point

iteration, is a very efficient batch algorithm that can be used to maximize both one-unit contrast functions and multi-unit contrast functions, including likelihood.

3.3 Fast ICA Algorithm

To begin with, we shall show the one-unit version of FastICA. By a "unit" we refer to a computational unit, eventually an artificial neuron, having a weight vector **w** that the neuron is able to update by a learning rule. The Fast ICA learning rule finds a direction, i.e. a unit vector **w** such that the projection **w***T* **x** maximizes nongaussianity. Nongaussianity is here measured by the approximation of negentropy $J(\mathbf{w}^T\mathbf{x})$ given in negentropy equ. Recall that the variance of **w***T* **x** must here be constrained to unity; for whitened data this is equivalent to constraining the norm of **w** to be unity.
The FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of **w***T* **x**. It can be also derived as an approximative Newton iteration (Hyvärinen, 1999a). Denote by *g* the derivative of the nonquadratic function *G* used.
for example the derivatives of the functions G are:

$$g_1(u) = \tanh(a_1 u),$$
$$g_2(u) = u\exp(-u^2/2)$$

where 1<a1<2 is some suitable constant, often taken as $a1 = 1$. The basic form of the FastICA algorithm is as follows

1. Choose an initial (e.g. random) weight vector $\mathbf{w}$.
2. Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}^+/\|\mathbf{w}^+\|$
4. If not converged, go back to 2.

Note that convergence means that the old and new values of **w** point in the same direction, i.e. their dot-product is (almost) equal to 1. It is not necessary that the vector converges to a single point, since **w** and **−w** define the same direction. This is again because the independent components can be defined only up to a multiplicative sign. Note also that it is here assumed that the data is prewhitened.

3.4 Fast ICA for several units

A simple way of achieving decorrelation is a deflation scheme based on a Gram-Schmidt-like decorrelation. This means that we estimate the independent components one by one. When we have estimated *p* independent components, or *p* vectors **w**1, ...,**w***p*, we run the one-unit fixed-point algorithm for **w***p*+1, and after every iteration step subtract from **w***p*+1 the "projections" **w***T p*+1**w***j***w***j* , *j* = 1, ..., *p* of the previously estimated *p* vectors, and then renormalize **w***p*+1:

$$\begin{aligned}&1.\ \text{Let } \mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \textstyle\sum_{j=1}^{p} \mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j \\ &2.\ \text{Let } \mathbf{w}_{p+1} = \mathbf{w}_{p+1} / \sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}\end{aligned}$$

$$\text{Let } \mathbf{W} = (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W}$$

where **W** is the matrix (**w**1, ...,**w***n*)*T* of the vectors, and the inverse square root (**WW***T*)−1/2 is obtained from the eigenvalue decomposition of **WW***T* = **FDF***T* as (**WW***T*)−1/2 = **FD**−1/2**F***T* . A simpler alternative is the following iterative algorithm .

$$\begin{aligned}&1.\ \text{Let } \mathbf{W} = \mathbf{W} / \sqrt{\|\mathbf{W}\mathbf{W}^T\|} \\ &\text{Repeat 2. until convergence:} \\ &2.\ \text{Let } \mathbf{W} = \tfrac{3}{2}\mathbf{W} - \tfrac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W}\end{aligned}$$

The normin step 1 can be almost any ordinarymatrix norm, e.g., the 2-normor the largest absolute row (or column) sum.

3.5 Properties of the FastICA Algorithm

1. The convergence is cubic (or at least quadratic), under the assumption of the ICA data model . This is in contrast to ordinary ICA algorithms based on (stochastic) gradient descent methods, where the convergence is only linear
2. Contrary to gradient-based algorithms, there are no step size parameters to choose. This means that the algorithm is easy to use.
3. The algorithm finds directly independent components of (practically) any non-Gaussian distribution using any nonlinearity g. This is in contrast to many algorithms, where some estimate of the probability distribution function has to be first available, and the nonlinearity must be chosen accordingly.
4. The performance of the method can be optimized by choosing a suitable nonlinearity g. In particular, one can obtain algorithms that are robust and/or of minimum variance. In fact, the two nonlinearities have some optimal properties

3.6 Applications

3.6.1 General applications

>>Blind source separation.

>>Feature extraction.

>> Blind deconvolution.

>> Other applications

1. Blind source seperation

The classical application of the ICA model is blind source separation. In blind source separation, the observed values of $\mathbf{x}$ correspond to a realization of an m-dimensional discrete-time signal x(t), t=1,2,.... Then the independent components $s_i(t)$ are called source signals, which are usually original, uncorrupted signals or noise sources. A classical example of blind source separation is the cocktail party problem. Assume that

several people are speaking simultaneously in the same room, as in a cocktail party. Then the problem is to separate the voices of the different speakers, using recordings of several microphones in the room. In principle, this corresponds to the ICA data model, where $x_i(t)$is the recording of the i-th microphone, and the $s_i(t)$ are the waveforms of the voices.

A simple artificial illustration of blind source separation is given in Figures below .In this illustration, deterministic signals were used for purposes of illustration. However, the spectral properties of the signals are not used in the ICA framework, and thus the results would remain unchanged if the signals were simply (non-Gaussian) white noise.

2. Feature extraction

Another application of ICA is feature extraction.Then the columns of $\mathbf{A}$represent features, and s_i is the coefficient of the i-th feature in an observed data vector $\mathbf{x}$. The use of ICA for feature extraction is motivated by the theory of redundancy reduction

an essentially equivalent method based on sparse coding was applied for extraction of low-level features of natural image data. The results show that the extracted features correspond closely to those observed in the primary visual cortex. These results seem to be very robust, and have been later replicated by several other authors and methods. A systematical comparison between the ICA features and the properties of the simple cells in the macaque primary visual cortex was conducted in a paper, where the authors found a good match for most of the parameters, especially if video sequences were used instead of still images.

3. Blind deconolution

Blind deconvolution is different from the other techniques discussed in this Section in the sense that (in the very simplest case) were are dealing with one-dimensional time signals (or time series) instead of multidimensional data, though blind deconvolution can

also be extended to the multidimensional case. Blind deconvolution is an important research topic with a vast literature. We shall here describe only a special case of the problem that is closely connected to ours.

In blind deconvolution, a convolved version $x(t)$ of a scalar signal $s(t)$ is observed, without knowing the signal $s(t)$ or the convolution kernel. The problem is then to find a separating filter h so that $s(t)=h(t)*x(t)$.

The equalizer $h(t)$ is assumed to be a FIR filter of sufficient length, so that the truncation effects can be ignored. A special case of blind deconvolution that is especially interesting in our context is the case where it is assumed that the values of the signal $s(t)$ at two different points of time are statistically independent. Under certain assumptions, this problem can be solved by simply whitening the signal $x(t)$

4. Other Applications

Due to the close connection between ICA and projection pursuit on the one hand, and between ICA and factor analysis on the other, it should be possible to use ICA on many of the applications where projection pursuit and factor analysis are used. These include (exploratory) data analysis in such areas as economics, psychology, and other social sciences, as well as density estimation, and regression.

3.6.2 practical applications

1. Separation of Artifacts in MEG Data

Magnetoencephalography (MEG) is a noninvasive technique by which the activity or the cortical neurons can be measured with very good temporal resolution and moderate spatial resolution. When using a MEG record, as a research or clinical tool, the investigator may face a problem of extracting the essential features of the neuromagnetic

signals in the presence of artifacts. The amplitude of the disturbances may be higher than that of the brain signals, and the artifacts may resemble pathological signals in shape.

The MEG signals were recorded in a magnetically shielded room with a 122-channel whole-scalp Neuromag- 122 neuromagnetometer. This device collects data at 61 locations over the scalp, using orthogonal double-loop pick-up coils that couple strongly to a local source just underneath. The test person was asked to blink and make horizontal saccades, in order to produce typical ocular (eye) artifacts. Moreover, to produce myographic (muscle) artifacts, the subject was asked to bite his teeth for as long as 20 seconds. Yet another artifact was created by placing a digital watch one meter away from the helmet into the shielded room.

2 Finding Hidden Factors in Financial Data

The assumption of having some underlying independent components in this specific application may not be unrealistic. For example, factors like seasonal variations due to holidays and annual variations, and factors having a sudden effect on the purchasing power of the customers like prize changes of various commodities, can be expected to have an effect on all the retail stores, and such factors can be assumed to be roughly independent of each other.

The factors have clearly different interpretations. The upmost two factors follow the sudden changes that are caused by holidays etc.; the most prominent example is the Christmas time. The factor on the bottom row, on the other hand, reflects the slower seasonal variation, with the effect of the summer holidays clearly visible. The factor on the third row could represent a still slower variation, something resembling a trend. The last factor, on the fourth row, is different from the others; it might be that this factor follows mostly the relative competitive position of the retail chain with respect to its competitors, but other interpretations are also possible.

3 Reducing Noise in Natural Images

The sample windows were taken at random locations. The 2-D structure of the windows

is of no significance here: row by row scanning was used to turn a square image window into a vector of pixel values. The independent components of such image windows are represented in Fig. 4. Each window in this Figure corresponds to one of the columns **a***i* of the mixing matrix **A**. Thus an observed image window is a superposition of these windows , with independent coefficients Now, suppose a noisy image model holds

$$z=x+n.$$

where **n** is uncorrelated noise, with elements indexed in the image window in the same way as **x**, and **z** is the measured image window corrupted with noise. Let us further assume that **n** is Gaussian and **x** is non-Gaussian.

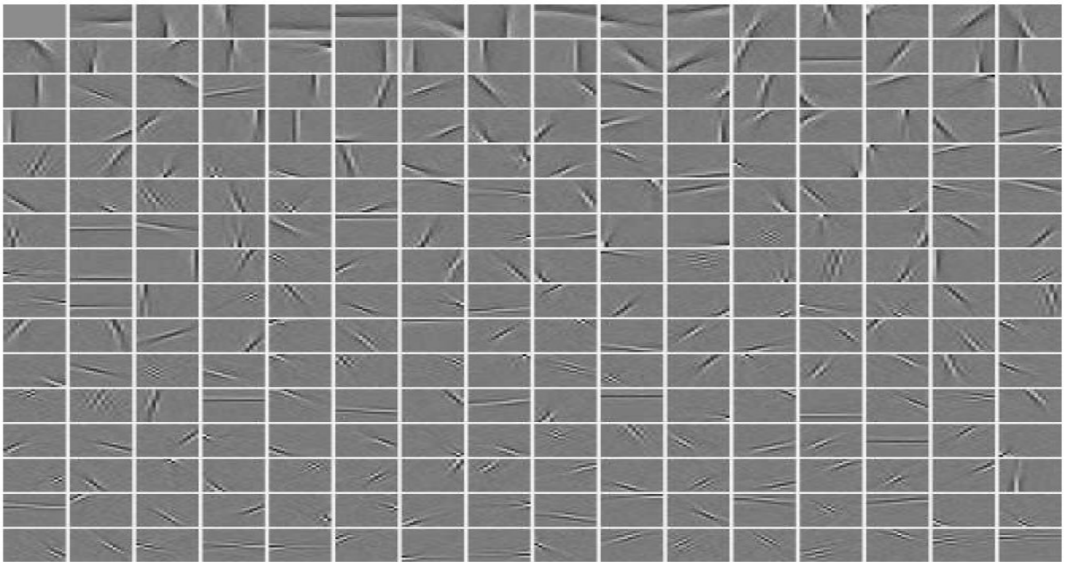


Fig .3.1 Basis functions in ICA of natural images. The input window size was 16×16 pixels. These basis functions can be considered as the independent features of images.

4.Telecommunications

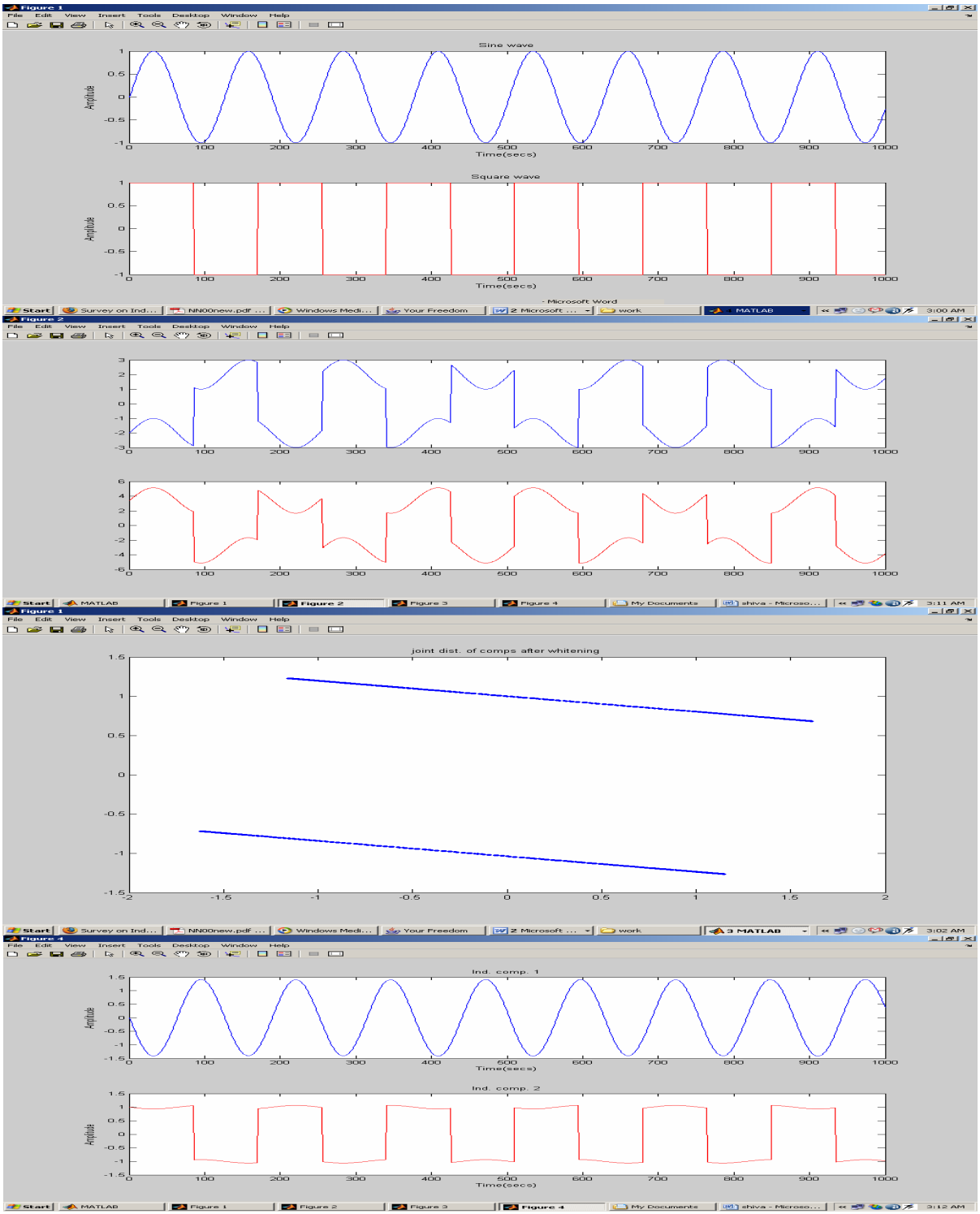
Finally, we mention another emerging application area of great potential: telecommunications. An example of a real-world communications application where blind separation techniques are useful is the separation of the user's own signal from the interfering other users' signals in CDMA (Code-Division Multiple Access) mobile Communications.

3.6.3 MATLAB SIMULATIONS FOR ICA USING VARIOUS ALGORTHEMS USING DIFFERENT NONLINEAR FUNCTIONS

PROGRAM --1

```
close all;
clear all;
max_iteration=10;
epsilon=0.00001;
n=2;          % no of sources
T=1000;       % sample size
A = sin(linspace(0,50, 1000));   % A
B = square(linspace(0,37, 1000)+5); % B
figure;
subplot(2,1,1); plot(A); % plot A
title('Sine wave'),xlabel('Time(secs)'),ylabel('Amplitude')
subplot(2,1,2); plot(B, 'r');    % plot B
title('Square wave'),xlabel('Time(secs)'),ylabel('Amplitude')
M1 = A - 2*B;                % mixing 1
M2 = 1.73*A+3.41*B;          % mixing 2
figure;
subplot(2,1,1); plot(M1);      % plot mixing 1
subplot(2,1,2); plot(M2, 'r'); % plot mixing 2
x = [M1;M2];
 [E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));        % inverse of square root
mx=mean(x');             % mean
xx=x-mx'*ones(1,T); % subtract the mean sample size=1000
xx=sq*E'*xx;
cov(xx')                 % the covariance is now a diagonal matrix
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
B=zeros(2);
```

```
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
    for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
        if  norm(w-w_old)<epsilon | norm(w+w_old)<epsilon
            B(:,i)=w;
            W(i,:)=w'*(sq*E');
            break;        end;
        w_old=w;
        u=xx'*w;
        umax= max(u)
     for k=1:T
         u1(k,1)=tanh(u(k,1));
         u2(k,1)=1-tanh(u(k,1))^2;
     end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
        w=w/norm(w)
    end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude'
```



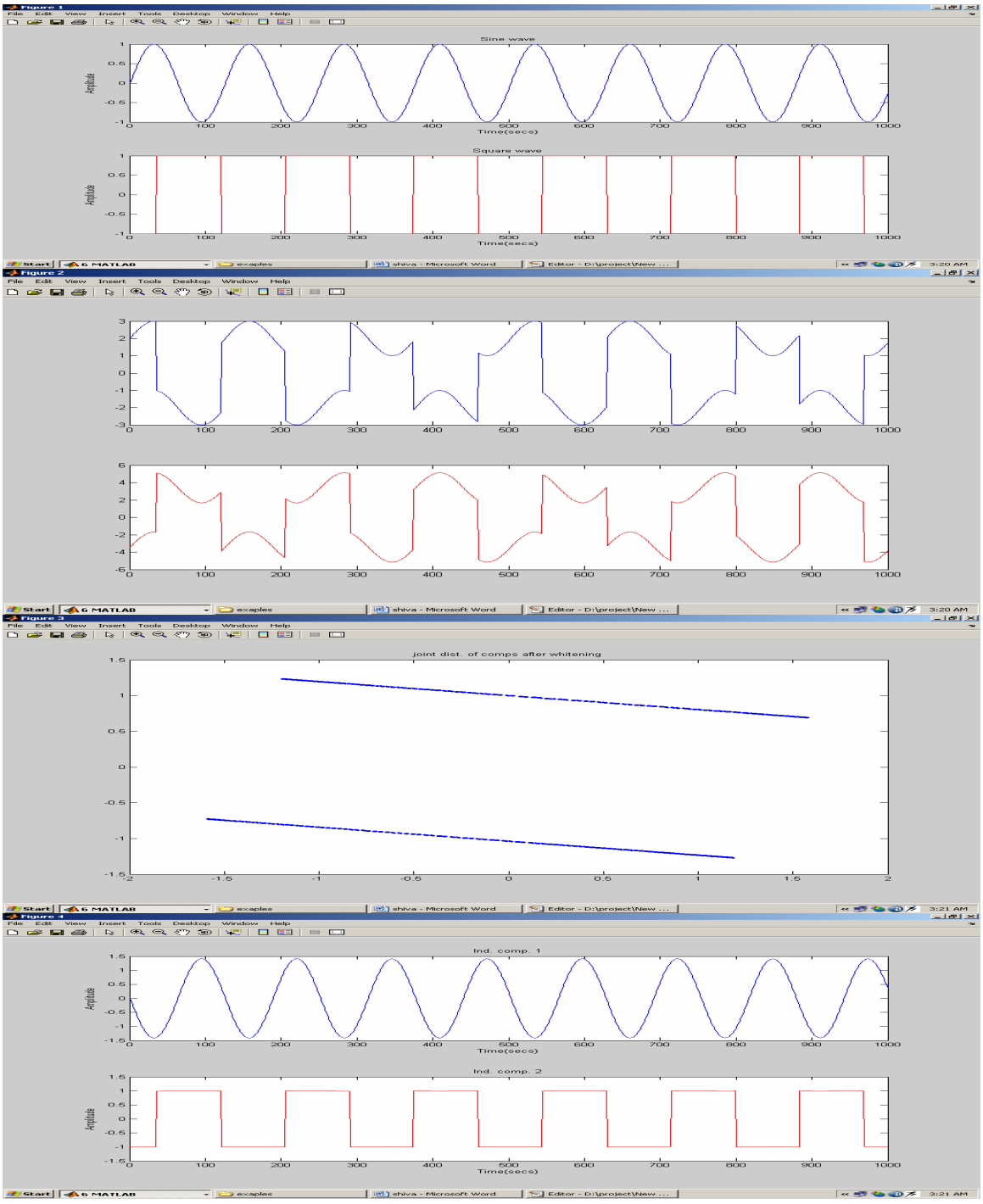
The estimates of the original source signals, estimated using only the observed signals in Fig. 3.1. The original signals were very accurately estimated, up to multiplicative signs. The four graphs represent ,the source signals ,the mixed signals ,joint distribution of the mixed signals, the recovered signals(outputs)

PROGRAM--2

```
close all;
clear all;
max_iteration=10;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = square(linspace(0,40, 10000));
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('Square wave'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
```

```
B=zeros(2);
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=u(k,1)*exp(-u(k,1)^2/2);
      u2(k,1)=(1-u(k,1)^2)*exp(-u(k,1)^2/2);
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');
```



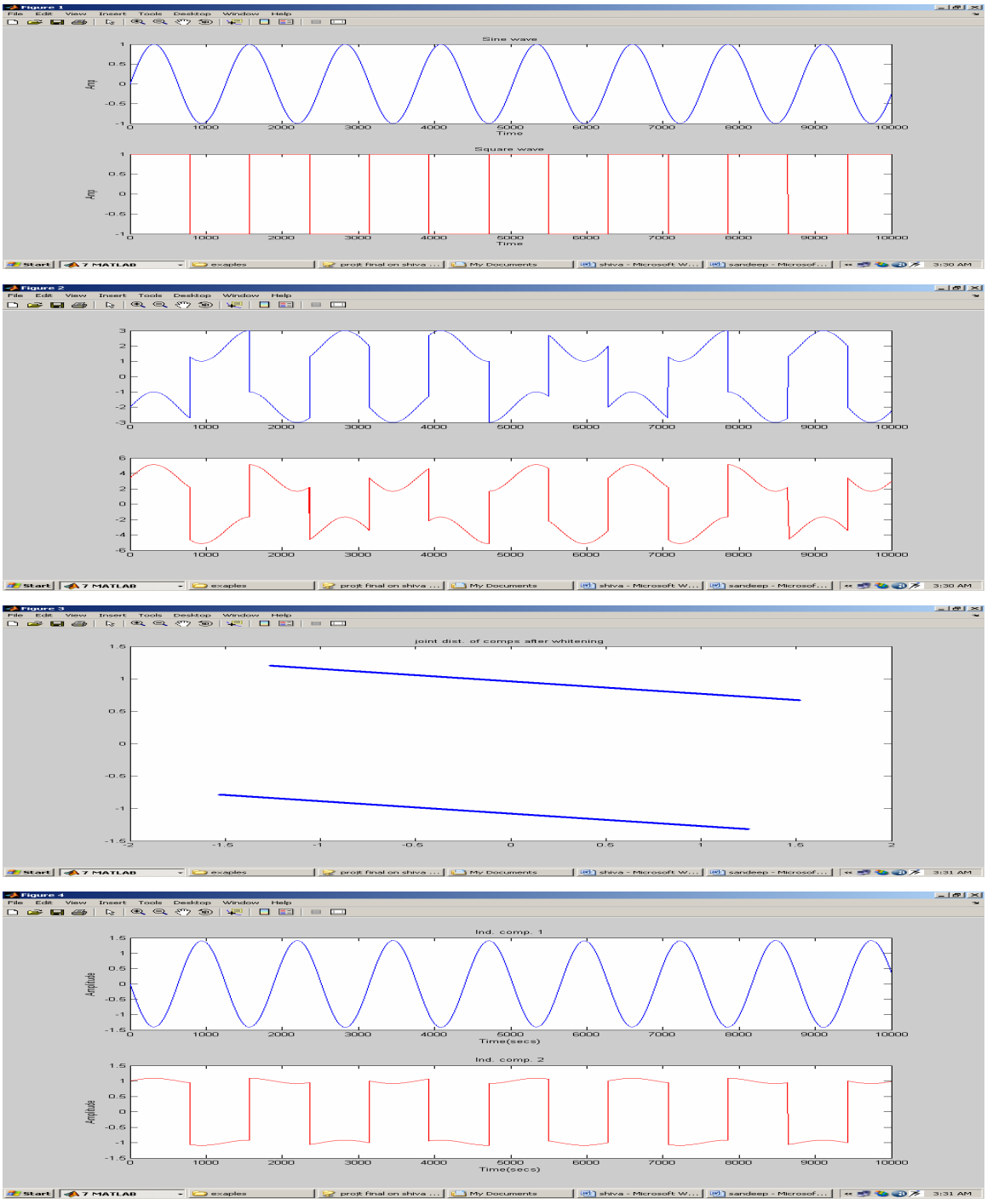
The estimates of the original source signals, estimated using only the observed signals in Fig. 3.2. Theoriginal signals were very accurately estimated, up to multiplicative signs The four graphs represent ,the source signals ,the mixed signals ,joint distribution of the mixed signals, the recovered signals(outputs)

PROGRAM -3

```
close all;
clear all;
max_iteration=10;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = square(linspace(0,40, 10000));
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('Square wave'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
```

```
B=zeros(2);
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=u(k,1)^3;
      u2(k,1)=3*u(k,1)^2;
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');
```



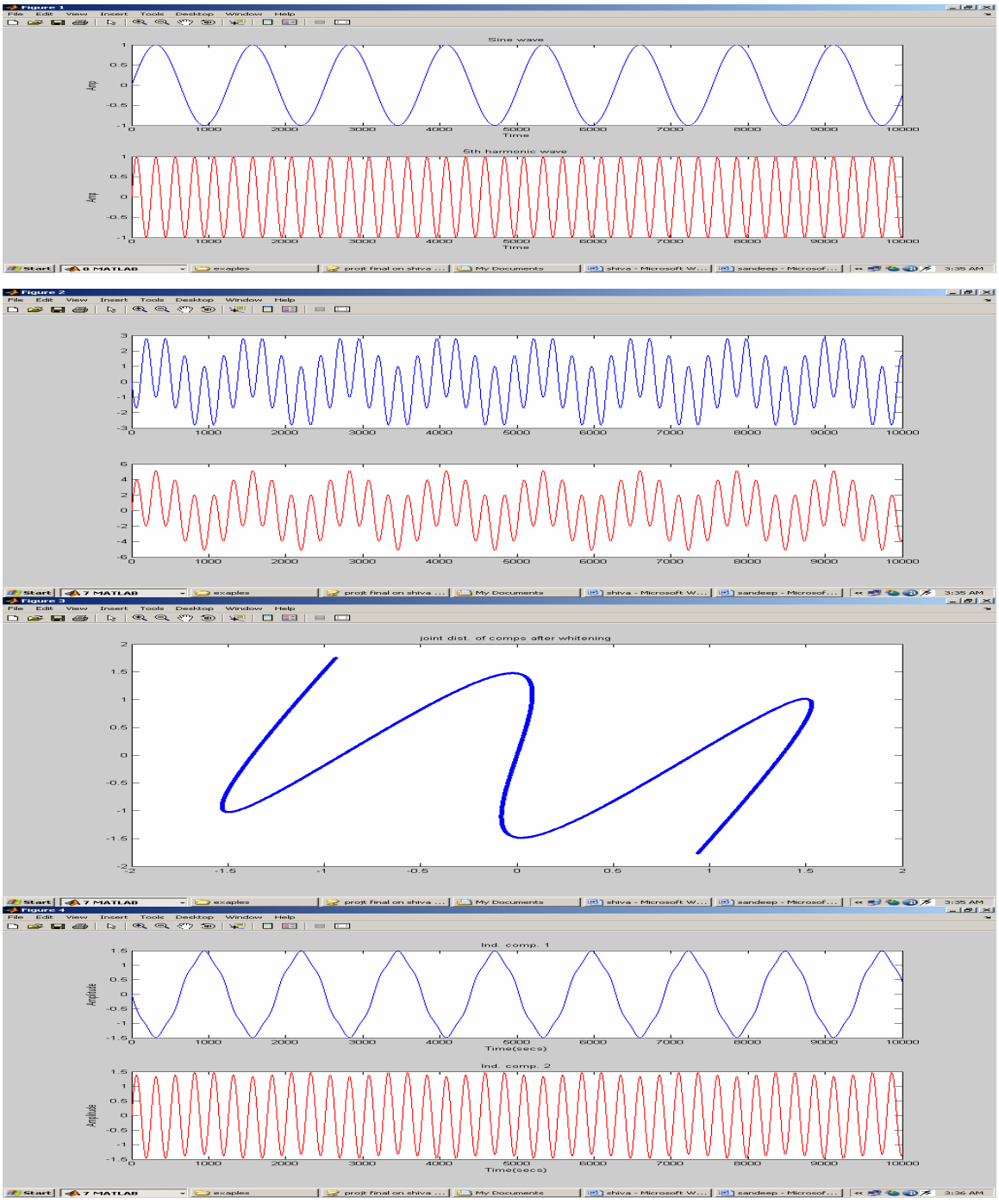
The estimates of the original source signals, estimated using only the observed signals in Fig. 3.3. Theoriginal signals were very accurately estimated, up to multiplicative signs The four graphs represent ,the source signals ,the mixed signals ,joint distribution of the mixed signals, the recovered signals(outputs)

Sine And its Hormonics-algo1

```
close all;
clear all;
max_iteration=100;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = sin(5*linspace(0,50, 10000));
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('5th harmonic wave'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
```

```
B=zeros(2);
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=tanh(u(k,1));
      u2(k,1)=1-tanh(u(k,1))^2;
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');
```



The estimates of the original source signals, estimated using only the observed signals in Fig. 3.4. Theoriginal signals were very accurately estimated, up to multiplicative signs The four graphs represent ,the source signals ,the mixed signals ,joint distribution of the mixed signals, the recovered signals(outputs)

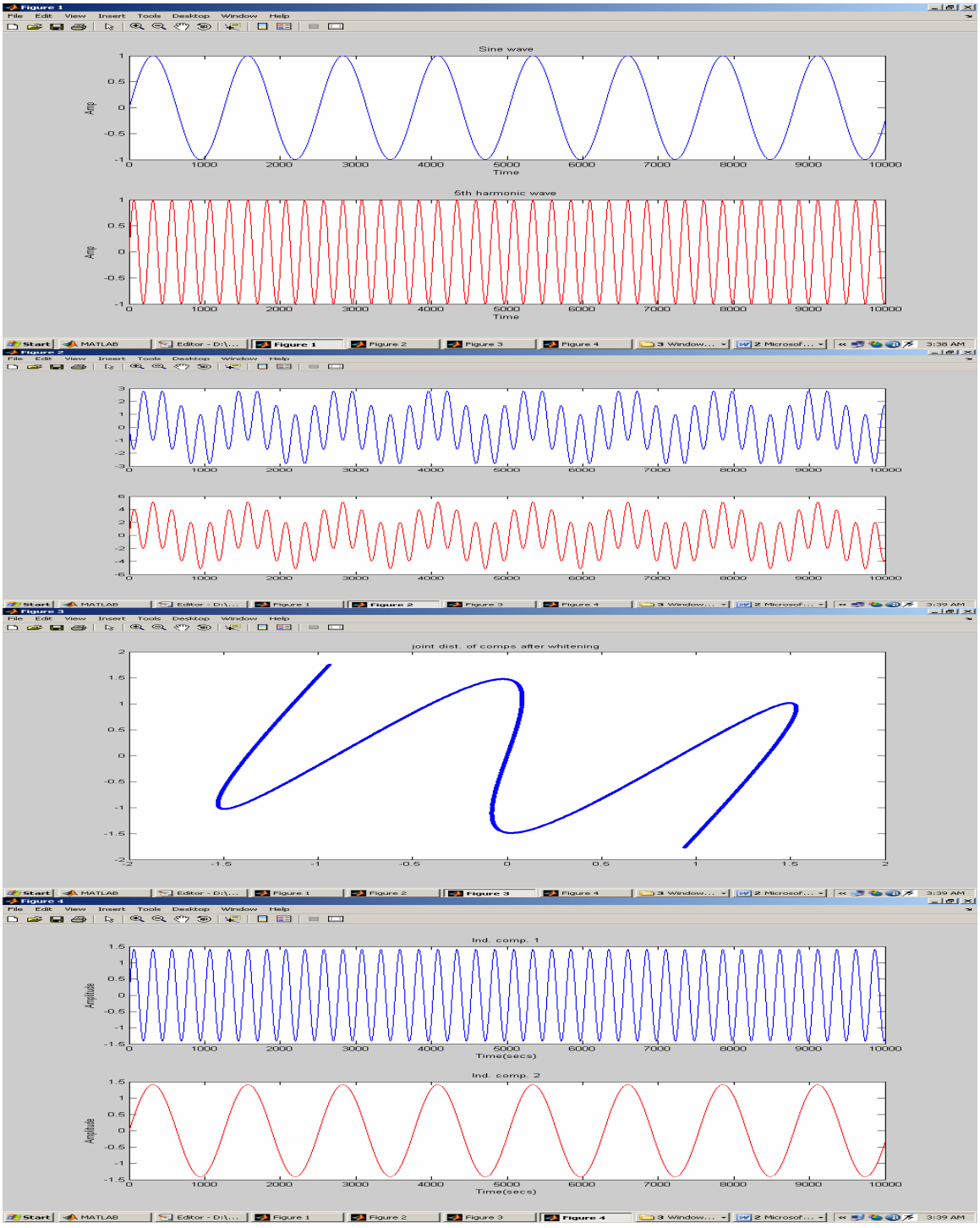
Sine And its Hormonis algo 3

```
close all;
clear all;
max_iteration=100;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = sin(5*linspace(0,50, 10000));
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('5th harmonic wave'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');

B=zeros(2);
```

```
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=u(k,1)^3;
      u2(k,1)=3*u(k,1)^2;
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');
```



The estimates of the original source signals, estimated using only the observed signals in Fig. 3.5. The original signals were very accurately estimated, up to multiplicative signs

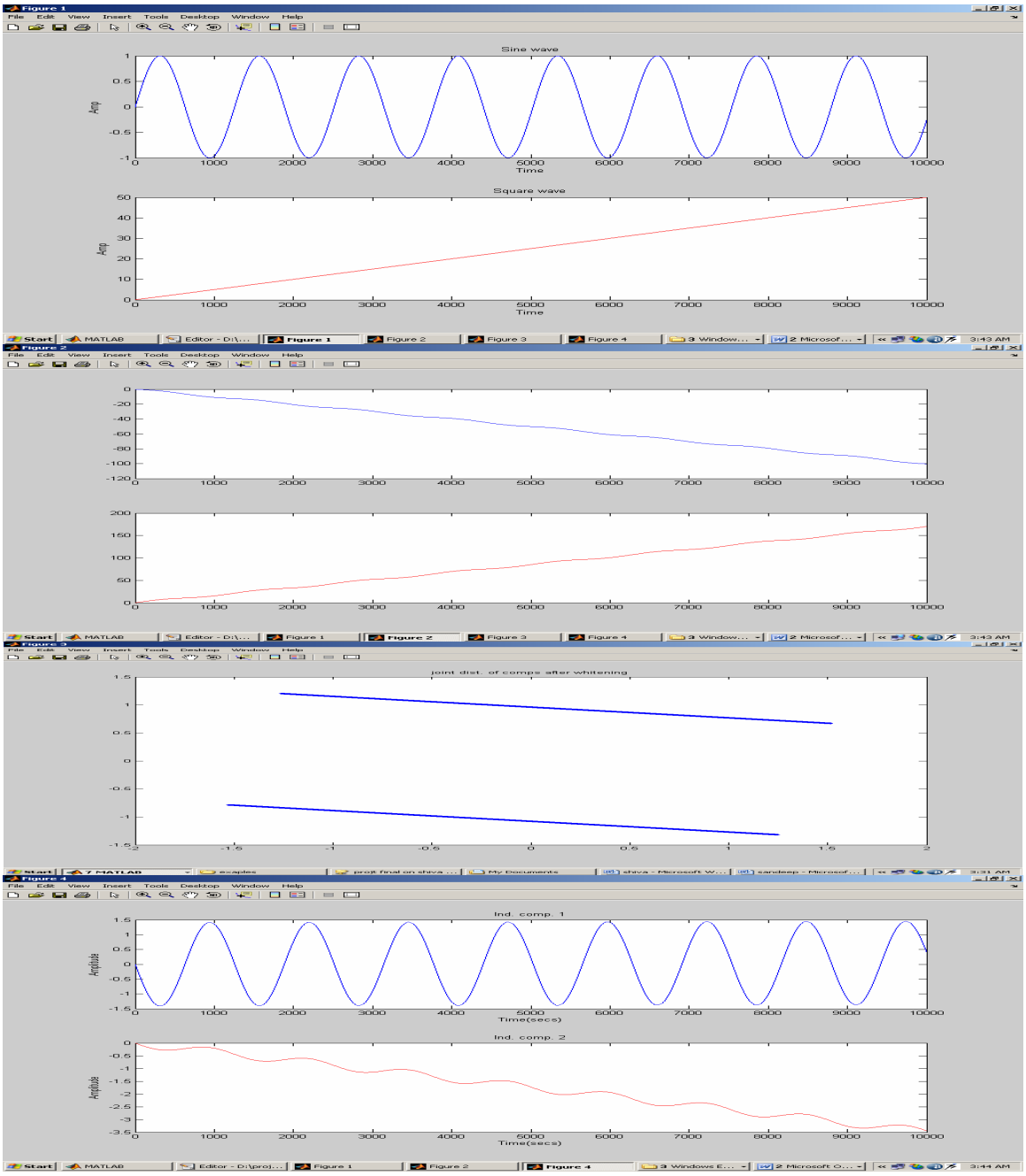
The four graphs represent ,the source signals ,the mixed signals ,joint distribution of the mixed signals, the recovered signals(outputs)

Sine and linear –algo2

```
close all;
clear all;
max_iteration=10;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = linspace(0,50, 10000);
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('Square wave'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
```

```
B=zeros(2);
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=u(k,1)*exp(-u(k,1)^2/2);
      u2(k,1)=(1-u(k,1)^2)*exp(-u(k,1)^2/2);
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');
```



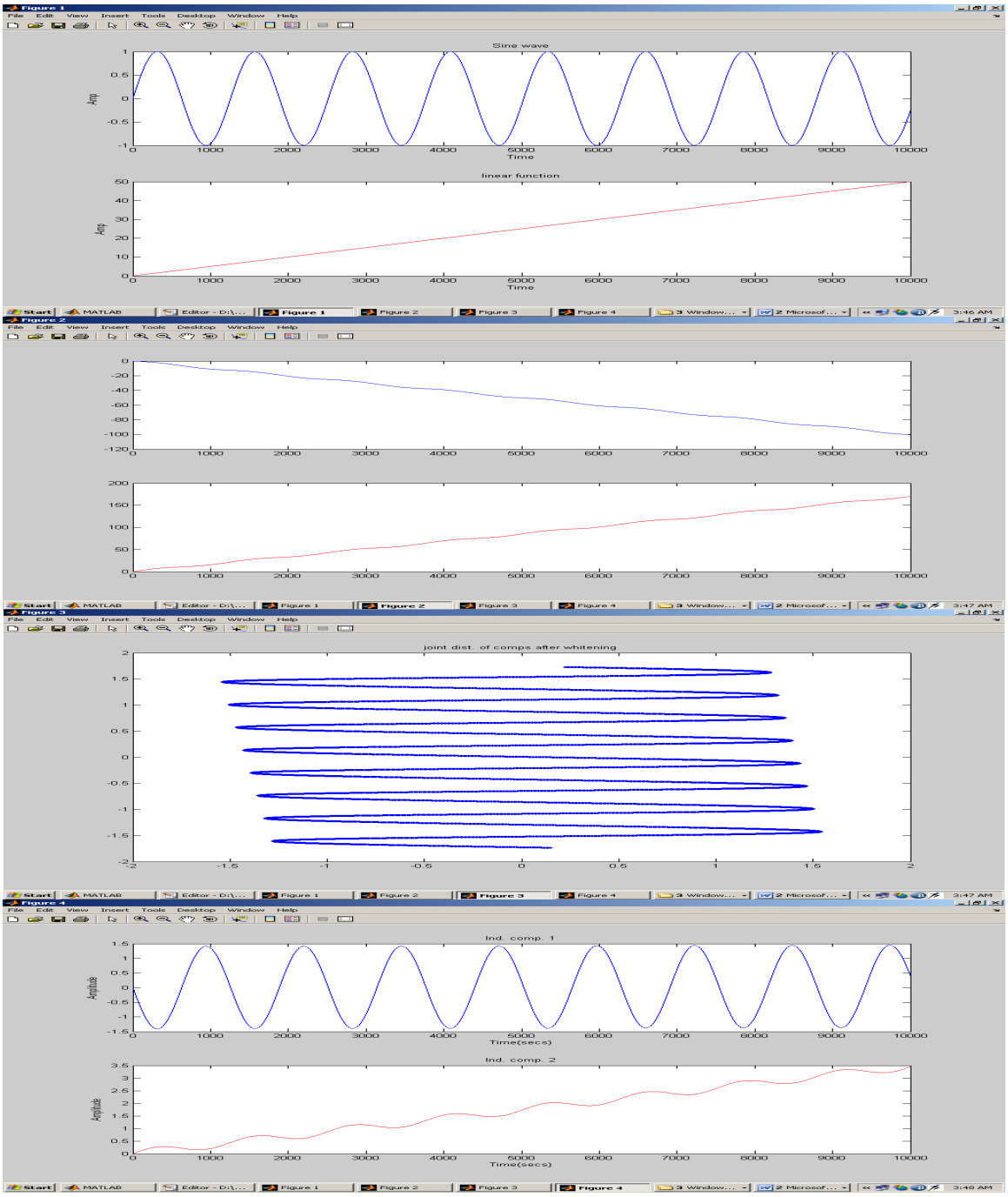
The estimates of the original source signals, estimated using only the observed signals in Fig. 3.6. The original signals were very accurately estimated, up to multiplicative signs The four graphs represent, the source signals, the mixed signals, joint distribution of the mixed signals, the recovered signals(outputs)

Sine And linear-algo 1

```
close all;
clear all;
max_iteration=10000;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = linspace(0,50, 10000);
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('linear function'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
```

```
B=zeros(2);
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=tanh(u(k,1));
      u2(k,1)=1-tanh(u(k,1))^2;
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');
```



The estimates of the original source signals, estimated using only the observed signals in Fig. 3.7. The original signals were very accurately estimated, up to multiplicative signs The four graphs represent, the source signals, the mixed signals, joint distribution of the mixed signals, the recovered signals(outputs)

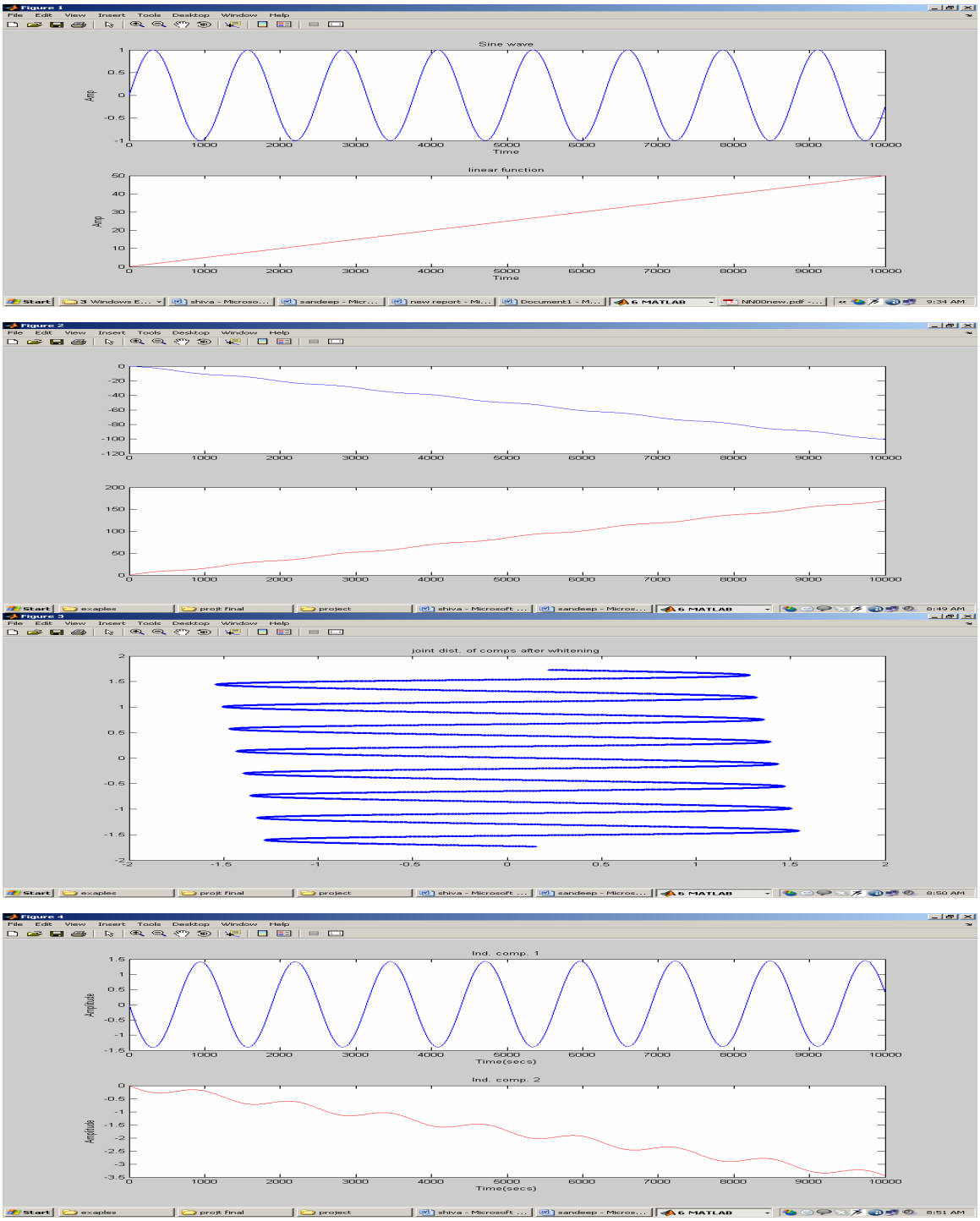
Sine and linear –algo-3

```
close all;
clear all;
max_iteration=10000;
converging_factor =0.00001;
n=2;
T=10000;
A = sin(linspace(0,50, 10000));
B = linspace(0,50, 10000);
figure;
subplot(2,1,1); plot(A);
title('Sine wave'),xlabel('Time'),ylabel('Amp')
subplot(2,1,2); plot(B, 'r');
title('linear function'),xlabel('Time'),ylabel('Amp')
M1 = A - 2*B;
M2 = 1.73*A+3.41*B;
figure;
subplot(2,1,1); plot(M1);
subplot(2,1,2); plot(M2, 'r');

x = [M1;M2];
[E,c]=eig(cov(x',1))
sq=inv(sqrtm(c));
mx=mean(x');
xx=x-mx'*ones(1,T);
xx=sq*E'*xx;
cov(xx')
figure; plot(xx(1,:), xx(2,:), '.');
title('joint dist. of comps after whitening');
```

```
B=zeros(2);
for i=1:2
    w=rand(2, 1)-0.5;
    w=w-B*B'*w;
    w=w/norm(w);
    w_old=zeros(size(w));
for j=1:max_iteration
        w=w-B*B'*w;
        w=w/norm(w);
 if  norm(w-w_old)<converging_factor | norm(w+w_old)<converging_factor
         B(:,i)=w;
         W(i,:)=w'*(sq*E');
         break;
  end;
      w_old=w;
      u=xx'*w;
      umax= max(u)
for k=1:T
      u1(k,1)=tanh(u(k,1));
      u2(k,1)=1-tanh(u(k,1))^2;
   end;
       w=(xx*u1)/size(xx,2)-(mean(u2))*w;
       w=w/norm(w)
   end
end
output=W*x;
figure;
subplot(2,1,1),plot(output(1,:))
title('Ind. comp. 1'),xlabel('Time(secs)'),ylabel('Amplitude');
subplot(2,1,2),plot(output(2,:),'r')
```

title('Ind. comp. 2'),xlabel('Time(secs)'),ylabel('Amplitude');



The estimates of the original source signals, estimated using only the observed signals in Fig. 3.8. The original signals were very accurately estimated, up to multiplicative signs The four graphs represent ,the source signals ,the mixed signals ,joint distribution of the mixed signals, the recovered signals(outputs)

Chapter 4

RESULTS AND DISCUSSION

4.1 Program 1

We have taken 1000 samples of sine and square wave of different frequencies ,if we mix these to signals in a random proportion to get the mixed signals using the gradient algorithm we are able to get back the original signals

The non linear functions used were :

1.tanh(u)

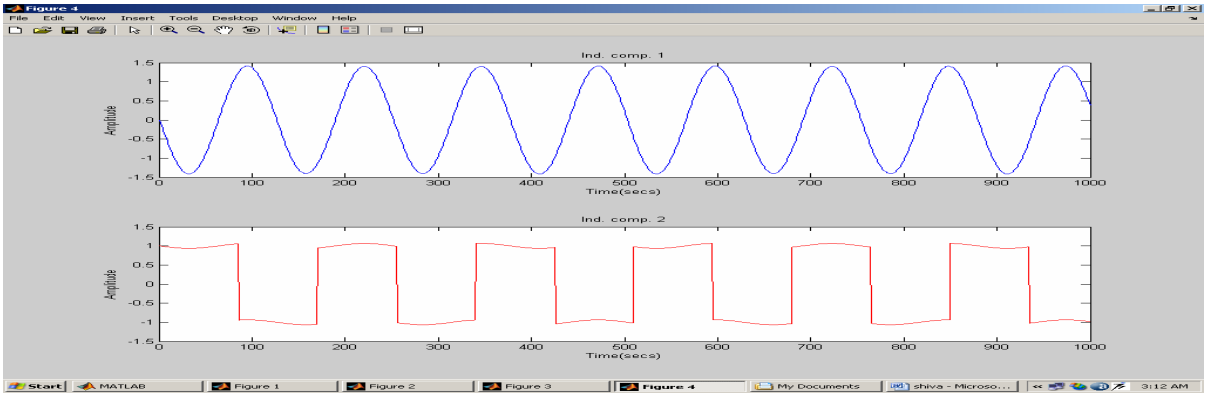
2. $1-\tanh^2(u)$

The error between original and the reconstructed signal decreases by increasing the number of iterations and by decreasing the converging factor (of order 0.000001) we reached the local maxima in the optimization landscape.

But by increasing number of iterations and decreasing the converging factor the computational complexity readily increases.

The order in which the outputs were recovered could not be predicted.

We could not determine the variances of source signals so we assumed each variable to be of unit variance.



4.2 Program 2

We have taken 1000 samples of sine and square wave of different frequencies ,if we mix these to signals in a random proportion to get the mixed signals using the gradient algorithm we are able to get back the original signals

The non linear functions used were :

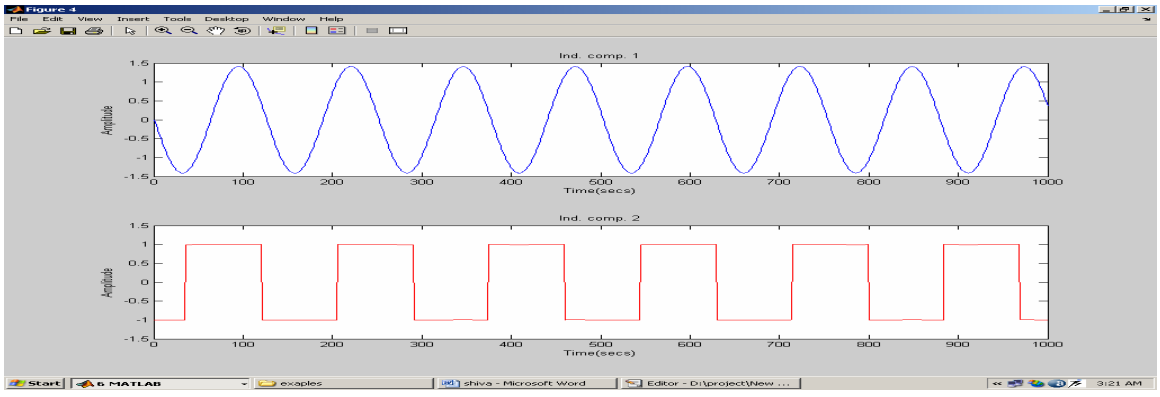
1. u(k)* exp(-u(k)^2/2)
2. (1- u(k))* exp(-u(k)^2/2)

The error between original and the reconstructed signal decreases by increasing the number of iterations and by decreasing the converging factor (of order 0.000001) we reached the local maxima in the optimization landscape.

But by increasing number of iterations and decreasing the converging factor the computational complexity readily increases.

The order in which the outputs were recovered could not be predicted.

We could not determine the variances of source signals so we assumed each variable to be of unit variance.



4.3 Program 3

We have taken 1000 samples of sine and its harmonic of different frequencies ,if we mix these to signals in a random proportion to get the mixed signals using the gradient algorithm we are able to get back the original signals

The non linear functions used were :

2.tanh(u)

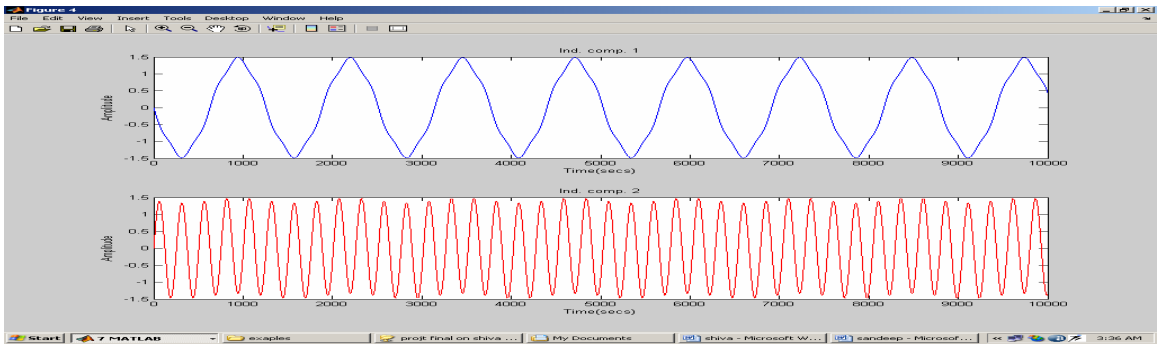
3.1-tanh(u)

The error between original and the reconstructed signal decreases by increasing the number of iterations and by decreasing the converging factor (of order 0.000001) we reached the local maxima in the optimization landscape.

But by increasing number of iterations and decreasing the converging factor the computational complexity readily increases.

The order in which the outputs were recovered could not be predicted.

We could not determine the variances of source signals so we assumed each variable to be of unit variance.



4.4 Program 4

We have taken 1000 samples of sine and harmonics ,if we mix these to signals in a random proportion to get the mixed signals using the gradient algorithm we are able to get back the original signals

The non linear functions used were :

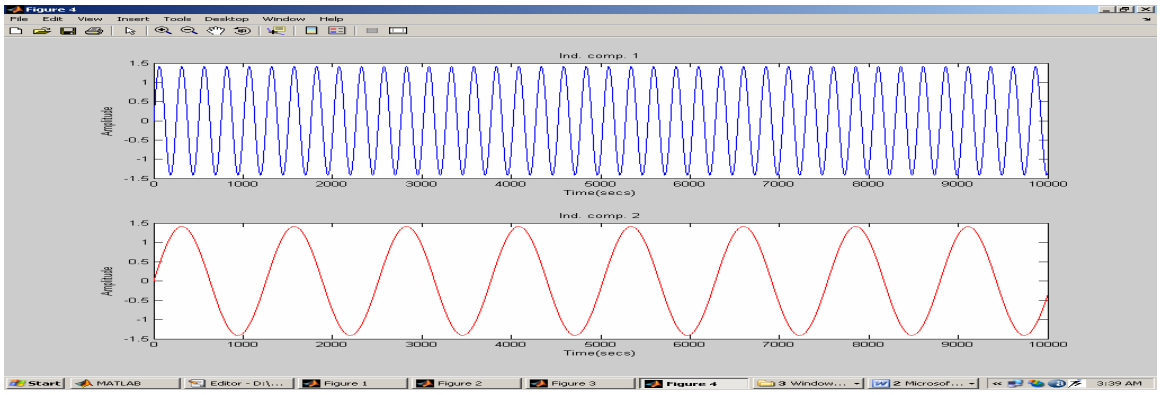
1. u(k)* exp(-u(k)^2/2)
2. (1- u(k))* exp(-u(k)^2/2)

The error between original and the reconstructed signal decreases by increasing the number of iterations and by decreasing the converging factor (of order 0.000001) we reached the local maxima in the optimization landscape.

But by increasing number of iterations and decreasing the converging factor the computational complexity readily increases.

The order in which the outputs were recovered could not be predicted.

We could not determine the variances of source signals so we assumed each variable to be of unit variance.



4.5 Program 5

We have taken 1000 samples of sine and linear function ,if we mix these to signals in a random proportion to get the mixed signals using the gradient algorithm we are able to get back the original signals

The non linear functions used were :

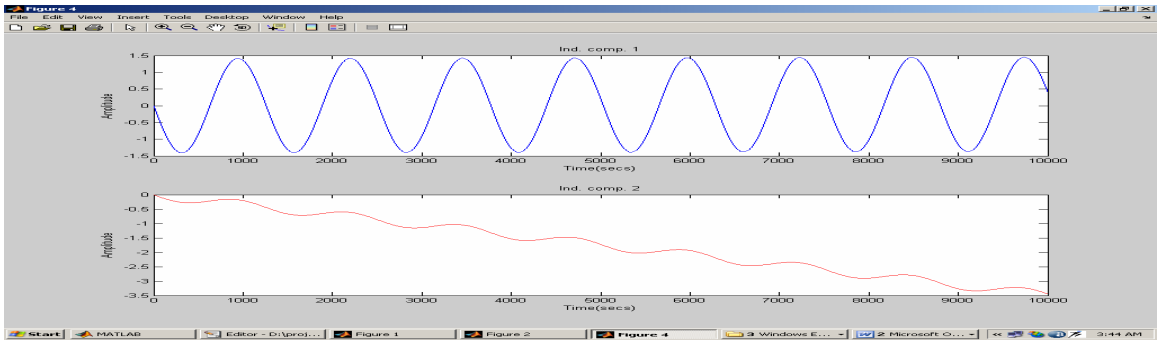
1. u(k)*exp(-u(k)^2/2);
2. (1-u(k,1)^2)*exp(-u(k,1)^2/2)

The error between original and the reconstructed signal decreases by increasing the number of iterations and by decreasing the converging factor (of order 0.000001) we reached the local maxima in the optimization landscape.

But by increasing number of iterations and decreasing the converging factor the computational complexity readily increases.

The order in which the outputs were recovered could not be predicted.

We could not determine the variances of source signals so we assumed each variable to be of unit variance.



4.6 Program 6

We have taken 1000 samples of sine and linear function ,if we mix these to signals in a random proportion to get the mixed signals using the gradient algorithm we are able to get back the original signals

The non linear functions used were :

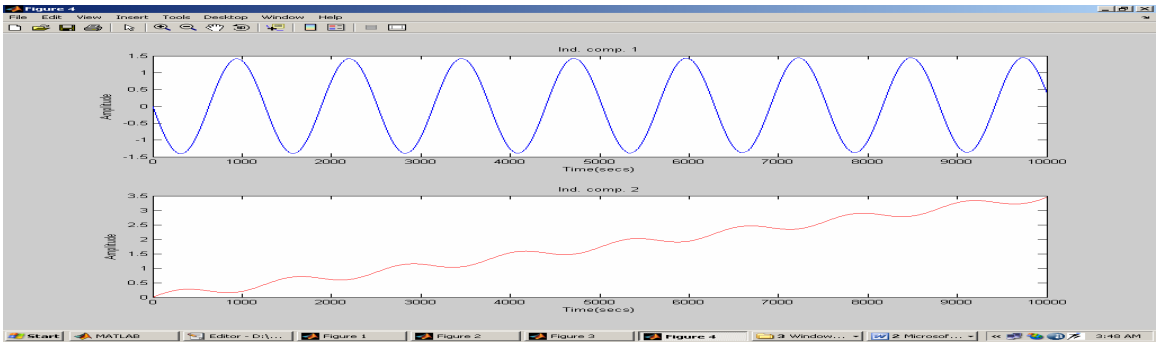
1. tanh(u)
2. $1-\tanh^2(u)$

The error between original and the reconstructed signal decreases by increasing the number of iterations and by decreasing the converging factor (of order 0.000001) we reached the local maxima in the optimization landscape.

But by increasing number of iterations and decreasing the converging factor the computational complexity readily increases.

The order in which the outputs were recovered could not be predicted.

We could not determine the variances of source signals so we assumed each variable to be of unit variance.



Chapter 5

CONLUSION AND REFERENSES:

5.1 Conclusion:

ICA is a very general-purpose statistical technique in which observed random data are linearly transformed into components that are maximally independent from each other, and simultaneously have “interesting” distributions. ICA can be formulated as the estimation of a latent variable model. The intuitive notion of maximum nongaussianity can be used to derive different objective functions whose optimization enables the estimation of the ICA model. Alternatively, one may use more classical notions like maximum likelihood estimation or minimization of mutual information to estimate ICA; somewhat surprisingly, these approaches are (approximatively) equivalent. A computationally very efficient method performing the actual estimation is given by the FastICA algorithm. Applications of ICA can be found in many different areas such as audio processing, biomedical signal processing, image processing, telecommunications, and econometrics.

Eg:

>>Blind source separation.

>>Feature extraction.

>> Blind deconvolution.

>> Separation of Artifacts in MEG Data

>>Finding Hidden Factors in Financial Data

>>Reducing Noise in Natural Images

>>Telecommunications

5.2 REFERENCES:

1."INDEPENDENT COMPONENT ANALYSIS"(book) by Aapo Hyvarinen, Juha Karhunen and Erkki Oja.(source: central library)

2 Amari, S.-I., Cichocki, A., and Yang, H. (1996). A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems.*

*3.*Bell, A. and Sejnowski, T. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159.

4. Cichocki, A. and Unbehauen, R. (1996). Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Trans. on Circuits and Systems*

5. **http://www.cis.hut.fi/aapo/papers/NCS99web/ --'**A SURVEY ON INDEPENDENT COMPONENT ANALYSIS'